

Micro Systems

JOURNAL

SPECIAL GRAPHICS ISSUE

Integration Artistry

**Programming
for EGA
Primitives**

**Dumping Graphics
to a Laser
Printer**

A Peek at Windows Products:

- *The Whitewater Group's Actor*
- *Microsoft Windows/386*



0 38351 19902 9

07



FREE

Disk cache package
and Turbo EGA for
hot performance
in disk and
graphics!

CONVERT YOUR PC, XT OR AT INTO A HIGHER FORM OF LIFE!

386 MOTHERBOARDS FOR 386 SPEED

Don't let your PC give up the ghost — Hauppauge has just arrived with a new spark of life: the 386 MotherBoard.™ Far more advanced than an accelerator card, our line of MotherBoards grace your PC, PC/XT, PC/AT or compatible with speeds equal to the IBM PS2 Model 80. And faster. Because we've built in 1 Megabyte of high speed RAM and a 387 math coprocessor socket for speeds that make users humble with awe.

OS/2 Compatible. To ensure a long, fruitful life, our 386 MotherBoard is compatible with the PC/AT (BIOS and I/O) — so you can run the new generation of DOS, OS/2. Our Board also runs Windows/386, UNIX V and PC-MOS/386. For more power, you'll find 16-bit expansion slots that accommodate the latest I/O expansion cards. No 386 accelerator card gives you so much versatility. **Only our 386 MotherBoard gives your PC a future with unlimited possibilities!**

The Critics Applaud! *PC Magazine* awarded our Board "The Editor's Choice" for 386 Replacement Boards. *PC World* called it "the Upgrade Product of the Year."

Technical Features ■ 16 MHz 80386 ■ 1 Megabyte of 100 nsec 4-way interleaved RAM ■ PC/AT compatible I/O and BIOS for support of OS/2 ■ Six 8-bit expansion slots ■ two 16-bit expansion slots (four on 386 MotherBoard/AT) ■ One 32-bit expansion slot for up to 12 Megabytes of high speed memory ■ Battery-powered clock/calendar

386 MotherBoard/PC or MotherBoard/XT	\$1495
386 MotherBoard/AT	\$1595
32-bit RAM Board	
(2 Mbytes installed; up to 4 Mbytes)	\$795
16 MHz 80387 math coprocessor	\$695
16-bit combination hard disk/ floppy disk controller	\$245

For more information on our easy-to-install MotherBoard, call:
1 (800) 443-6284. In New York, call (516) 434-1600.

Hauppauge Computer Works, Inc.
175 Commerce Drive,
Hauppauge, New York 11788

Hauppauge!

New Prices

OS/2

WINDOWS FOR DATA®

MULTI-LEVEL
MENU SYSTEM

NESTED
POP-UP FORMS

SCROLLABLE
REGION

CHOICE LIST

CLOCK

POP-UP
WINDOW

RUNNING
TOTALS

MESSAGE
WINDOW

Invoices: Create Review Print Exit
INVOICE
 Invoice No.: 000784 Date: 12/03/87 Time: 16:43:15
 Customer:
 William Jones
 Innovative Software
 351 Bulletin Avenue
 Needham, MA 02194
 (617) 394-5512
 Search for customer record? (Y/N): N
 Enter customer information? (Y/N): N
 Enter billing address? (Y/N): N
 Enter marketing information? (Y/N): N

No.	PRODUCT	DESCRIPTION	QUANTITY	PRICE	AMOUNT
5	WDMS	Windows for Data Microsoft	10	295.00	2950.00
6	WDLA	Windows for Data Lattice	5	295.00	1475.00
7	WDTC	Windows for Data Turbo C	5	295.00	1475.00
8	WDXE	Windows for Data XENIX	2	795.00	1590.00
9			0	0.00	0.00

Subtotal: 11325.00
 Shipping: 0.00
TOTAL : 11325.00
 Payment : 0.00

WDXE Windows for Data - XENIX
 WDBB2 Windows for Data - 3B2 Unix
 WDSU2 Windows for Data - Sun Unix
 WDVMS Windows for Data - MicroVax
 WDVMS4 Windows for Data - Vax 780

Cursor keys scroll, ENTER selects and ESC exits choice menu

If you program in C, take a few moments to learn how Windows for Data can help you build a state-of-the-art user interface.

- ✓ Create and manage menus, data-entry forms, context-sensitive help, and text displays — all within windows.
- ✓ Develop window-based OS/2 programs right now, without the headaches of learning OS/2 screen management. Run the same source code in PC DOS and OS/2 protected mode.
- ✓ Build a better front end for any DBMS that has a C-language interface (most popular ones do).



NO WALLS

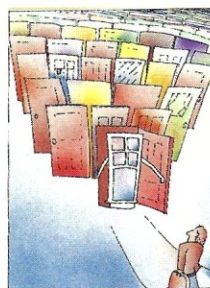
If you've been frustrated by the limitations of other screen utilities, don't be discouraged. You won't run into walls with Windows for Data. Our customers repeatedly tell us how they've used our system in ways we never imagined — but which we anticipated by designing Windows for Data for unprecedented adaptability. You will be amazed at what you can do with Windows for Data.

FROM END TO BEGINNING

Windows for Data begins where other screen packages end, with special features like nested pop-up forms and menus, field entry from lists of choices, scrollable regions for the entry of variable numbers of line items, and an exclusive built-in debugging system.

YOU ARE ALWAYS IN CHARGE

Control functions that you write and attach to fields and/or keys can read, compare, validate, and change the data values in all fields of the form. Upon entry or exit from any field, control functions can call up subsidiary forms and menus, change the active field, exit or abort the form, perform almost any task you can imagine.



versions are fully compatible with Microsoft Windows.
No royalties.

OUR WINDOWS WILL OPEN DOORS

Our windows will open doors to new markets for your software. High-performance, source-code-compatible versions of Windows for Data are now available for PC DOS, OS/2, XENIX, UNIX, and VMS. PC DOS

MONEY BACK GUARANTEE

You owe it to yourself and your programs to try Windows for Data. If not satisfied, you can return it for a full refund.

Prices: PC DOS \$295, Source \$295. OS/2 \$495. XENIX \$795. UNIX, VMS, please call.

Call: (802) 848-7731

Telex: 510-601-4160 VCSOFT

ext. 33

FAX 802-848-3502



**Vermont
Creative
Software**

21 Elm Ave.
Richford,
VT 05476



Finally, there's an SQL that gets back to BASIC. And COBOL. And C. And Pascal.

As a programmer, you've probably already faced it — the database dilemma. Do you use an SQL for easy database handling, or a true programming language for maximum power and flexibility?

Now you can do both with XQL®, the relational data management system from the developers of Btrieve.®

The Programmer's SQL. With XQL, you can access your data with the ease of Structured Query Language through simple subroutine calls from traditional programming languages. XQL supports standard SQL syntax, including subqueries, unions and security groups.

XQL Relational Primitive Operations.

In addition, XQL lets you bypass the SQL level and perform highly efficient, relational primitive operations directly. You get all the functionality of a relational database model without the constraints of a 4th generation language.

Building on Btrieve. The heart of Novell's family of data management tools is Btrieve. By letting you access multiple records at a time, XQL adds a powerful dimension to Btrieve. XQL incorporates sophisticated data manipulation features which

allow you to access data by field name, move forward or backwards through the database, compute fields from other fields or constants, and even work with composite records built from multiple, joined Btrieve files.

Like Btrieve, XQL offers features like multi-user support, fault tolerance, comprehensive documentation, and expert technical support. And you never pay royalties on your XQL applications.

Solve the database dilemma with XQL, the SQL that speaks your language. Only \$795.* See your Authorized Novell Gold Reseller, or call us at (512) 346-8380.

For more information, call from your modem 1-800-444-4472 (8 bit, no parity, 1 stop bit) and enter the access code NVXQL13.



For software solutions,
you should be seeing red.

*Suggested retail price (US dollars) ©1988 Novell Inc., World Headquarters, 122 East 1700 South, Provo, Utah 84601 (801) 379-5900
Requires Btrieve 4.x and PC-DOS or MS-DOS 2.x, 3.x.
Circle Reader Service Number 65

FEATURE ARTICLES

Assembly Language Primitives for EGA Graphics Under Turbo C

This article begins to lay the foundation for a general purpose graphics library written in C for the EGA. The cornerstone for this foundation—the graphic primitives—and are the most speed-critical, low-level routines.
 by Thomas Dwyer, Arthur Huang, and John King 20

A Routine to Dump Graphics Screens to Laser Printers

In this article, the author explains the inner workings of Hercules board graphics as they relate to the HP LaserJet printer. He also provides a memory-resident program that lets you dump any kind of Hercules-driven graphics to a LaserJet with the press of a hot key.
 by Jeffrey Ross 35

PRODUCT REVIEWS

Microsoft Windows/386

Why wait for OS/2's Presentation Manager? Windows/386 is available right now for running multiple DOS and Windows applications.
 by William Wong 29

Actor

If you are looking for an easy way to jump into Windows, this object-oriented programming language makes it possible for the uninitiated programmer to create his first Windows application in just a few days.
 by Daniel Feenberg, Ph.D. 44

Novell NetWare and Banyan VINES/286

Which operating system provides the most intelligent network alternative? When comparing NetWare to VINES, the big question is how many file servers do you need to support.
 by William Wong 50

VM/386

As the author explains, VM/386 acts as a multitasking "supervisor" to create multiple, independent tasks for 80386-based microcomputers running DOS.
 by Charles H. Strom 58

COLUMNS

From the Editor's Desk by Sol Libes

PS/2 MCA Versus the Enhanced AT Bus 6

The C Forum by Don Libes

Something for Everyone 14

LANscape by Patrick H. Corrigan

LANs on Telephone Wire 17

Scientific Computer User by A.G.W. Cameron

Upgrading to a Sun 4 60

In the Public Domain by Robert Blacher

Zmodem—Why a New File Transfer Protocol? 64



About the cover: Let's consider the integrator as artist. Not only is he responsible for designing elegant computer configurations, he is also charged with programming menus and developing user interfaces, helping users work with CAD/CAE, helping create presentation graphics, and solving a host of other graphic-related problems. And just as the artist needs brushes and canvas, the computer professional needs other tools, such as those explored in this issue.

Cover photograph by Michael Carr

DEPARTMENTS

News & Views..... 8

There Is Mail 12

Ad Index..... 40

New Products..... 68



DESQview API Reference Manual

This is the primary source of information about the DESQview API. It contains all you need to know to write assembly language programs that take full advantage of DESQview's capabilities. The Reference manual comes with an include file containing symbols and macros to aid you in development. **AVAILABLE NOW!**

DESQview API C Library

The DESQview API C Library provides C Language interfaces for the entire set of API functions. It supports the Lattice C, Metaware C, Microsoft C, and Turbo C compilers for all memory models. Included with the C Library

package is a copy of the API Reference Manual and source code for the library. **AVAILABLE NOW!**

DESQview API Debugger

The DESQview API Debugger is an interactive tool that enables the API programmer to trace and single step through API calls from several concurrently running DESQview-specific programs. Trace information is reported symbolically along with the program counter, registers, and stack at the time of the call. Trace conditions can be specified so that only those calls of interest are reported. **AVAILABLE JUNE 88**


DESQ
view
Quarterdeck

CIRCLE 86 ON READER SERVICE CARD

Introducing DESQview 2.0 API Tools

Bringing new power to DOS

DESQview API Panel Designer

The DESQview API Panel Designer is an interactive tool to aid you in designing windows, menus, help screens, error messages, and forms. It includes an editor that lets you construct an image of your panel using simple commands to enter, edit, copy, and move text as well as draw lines and boxes. You can then define the characteristics of the window that will contain the panel, such as its position, size, and title. Finally, you can specify the locations and types of fields in the panel.

The Panel Designer automatically generates all the DESQview API data streams necessary

to display and take input from your panel. These data streams may be grouped together into panel libraries and stored on disk or as part of your program. AVAILABLE JUNE 88

DESQview API Pulldown Menu Manager

The DESQview API Pulldown Menu Manager is an interactive tool to aid you in designing pulldown menus. This DESQview API tool assists you in giving your DOS program an OS/2-like look and feel. AVAILABLE JULY 88

MS-DOS and IBM PC-DOS are both trademarks of Microsoft Corporation and IBM Corporation respectively.

Quarterdeck Office Systems
150 Pico Boulevard
Santa Monica, CA 90405
(213) 392-9851



Call for registration information (213) 392-9851

From the Editor's Desk

PS/2 MCA Versus the Enhanced AT Bus

The PS/2 MicroChannel Architecture (MCA) has delivered new performance levels for IBM's 286- and 386-based systems. It provides a multimaster busing system that can support more intelligent, high-performance coprocessor cards with faster throughput.

But let's face it, IBM's main purpose in introducing a new personal computer bus design was not to improve performance and add features. Rather, it was to regain market share by preventing cloning.

The PS/2 models 25 and 30 are essentially repackaged XT's and do not include the MCA. They do offer the additional features of faster processing, VGA graphics, and 3¹/₂-inch floppy disk drives. However, PC/XT clones are now selling even better than before the 25 and 30 were introduced, because XT clone performance has improved.

The PS/2 models 50 and 60 are 286, MCA-based machines. But they also do not appear to be affecting sales of AT clones. In fact, 286 AT clones are selling even better than ever as clone makers improve their performance and add features. Most of the AT-compatibles being currently sold provide performance equal to or better than the PS/2 models 50 and 60, and at significantly lower cost. And it is becoming apparent that most of these systems will be upgradable to run OS/2, should one want to run the new operating system.

The clone builders are also readying clones of the models 30, 50 and 60. Some of these clones have already been demoed and shipments may begin late this year. But it is doubtful that there will be much of a demand for these clones. Their introduction will be more of a market research ploy on the part of the cloners rather than a serious attempt to compete against the IBM systems.

The 386 marketplace is a whole other story. Here Compaq has set the pace. Where once it was merely a cloner of IBM PC/XT/AT systems it has now become, with 386 systems, the technology leader and IBM is now a follower. Compaq introduced the first 386 system fully 10 months before IBM. Compaq introduced a dual bus architecture. It retained the standard 16-bit AT I/O bus for interfacing to peripherals and introduced a new 32-bit memory bus. This approach allowed compatibility with the large number of AT plug-in cards and, at the same time, improved processing performance via a high-speed memory bus.

IBM's PS/2 Model 80 386-based machine only equalled the performance of the Compaq machine. Shortly after IBM's introduction, Compaq introduced a new version of the DeskPro 386 with a 20-MHz 386 chip, improved memory caching, and an interface for the high-speed Weitek math coprocessor as well as the Intel 387 math coprocessor. IBM followed soon after with its 20-MHz unit that does not measure up to the Compaq unit in performance. AST has taken the same dual-bus approach, and other clone makers are expected to do likewise. And, AST has gone one big step further—it has developed a multimaster extension to the AT bus architecture. Further, it has promised to publish the specifications for this extension in the hope that it may become a standard the way its previous extended memory specification was adopted as the EMS 4.0 standard.

What Compaq and AST are demonstrating is that a dual-bus architecture consisting of the old AT 16-bit I/O bus and a well-designed 32-bit-wide memory bus, and extensions to the AT bus can provide an effective alternative to IBM's proprietary MicroChannel bus architecture.

The result is that IBM, which only a few years ago claimed a greater than 50 percent share of the personal computer market, has seen its share drop to less than 20 percent. The clone manufacturers, led by companies such as Compaq, AST, Tandy, and Zenith, are proving to be very nimble and aggressive marketers, as well as talented technology developers. IBM will have to come up with some scheme other than the PS/2 MCA and OS/2 if it expects to keep its market share from shrinking even further.

Sol Libes

Sol Libes
Editor and Founder

For the PC Systems Integrator

MicroSystems
JOURNAL

EDITORIAL

Founder and Editor Sol Libes
Technical Editors Stephen R. Davis
Don Libes
Associate Editors Lennie Libes
Susan Libes
Contributing Editors A.G.W. Cameron
Patrick Corrigan
P.L. Olympia
Managing Editor Thomas M. Woolf

PRODUCTION

Art & Production
Director Larry L. Clay
Art Director Kobi Morgan
Asst. Art Director Barbara Mautz
Typographer Lorraine Buckland

CIRCULATION

Director of Circulation Maureen Kaminski
Asst. Circulation Mgr. Andrea Weingart
Direct Mktg. Specialist Kathleen Shay
Fulfillment Coord. Francesca Martin
Newsstand Mgr. Sarah Frisbie

ADMINISTRATION

V.P. Finance & Operations Kate Wheat
Business Mgr. Betty Trickett
Accounting Supv. Mayda Lopez-Quintana
Accts. Payable Asst. Luanne Rocklewitz
Accts. Receivable Asst. Wendy Ho

ADVERTISING

Advertising Director Richard Mixer
National Account Mgr. Dwight Schwab
National Account Mgr. Tami Brenton
Advertising Coord. Shaun Hooper

M&T Publishing, Inc.

Chairman of the Board Otmir Weber
Director C.F. Von Quadt
President & Publisher Laird Foshay
V.P. of Publishing William P. Howard

Micro/Systems Journal (ISSN 8750-9482) is published monthly by M & T Publishing, Inc., 501 Galveston Drive, Redwood City, CA 94063; (415) 366-3600. Second-class postage paid at Redwood City and at additional entry points.

Article Submission: If you have a specific area of expertise or interest and would like to contribute, please write *Micro/Systems Journal*, P.O. Box 1192, Mountainside, NJ 07092; (201) 522-9347, or contact M & T Publishing, Inc., 501 Galveston Drive, Redwood City, CA 94063; (415) 366-3600. Please do not submit articles without first contacting the editors. Author's guidelines available upon request.

Correspondence: Please send letters to the editor to *Micro/Systems Journal*, 501 Galveston Drive, Redwood City, CA 94063. Other editorial correspondence may also be directed to P.O. Box 1192, Mountainside, NJ 07092. The editors may also be reached via MCI Mail (SLIBES or MSJ).

Advertising Rates: Available upon request. Call (415) 366-3600 or write to: Advertising Department, *Micro/Systems Journal*, 501 Galveston Drive, Redwood City, CA 94063.



VM/386™. The Fast Track To Multitasking.

Run smart. Run efficient. Run VM/386 on your 386-based PC and start multitasking now! No detours. no waiting, no runaround.

True multitasking. VM/386 uses the virtual 8086 mode of the 80386 processor to create many Virtual Machines (VMs) in one computer.

You can load a different application into each VM. Each VM has up to 640K RAM, plus its own DOS, CONFIG.SYS, AUTOEXEC.BAT and memory-resident programs along with its applications. Tailor each VM to your needs. You have complete control.

Each VM is protected from the others. A malfunction in one VM won't affect the others, but all VMs can share the same disk and other peripherals.

Recalculate a spreadsheet, sort a database file, and receive your E-mail—all at the same time. You can even work with two AutoCAD™ programs concurrently. EGA applications run perfectly too—background and foreground.

Protect your investment in software. No need to buy anything new. VM/386 runs existing DOS programs, unmodified. No PIF files required.

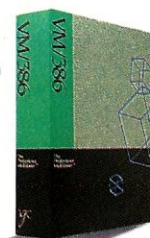
Eliminate the hidden costs of retraining. VM/386 is easy to install, easy to learn, and easy to use. There's no new operating system, interface, or application to learn.

Get on the fast track to multitasking. Call (408) 986-8373 for more information or to order VM/386. Everything else is just the runaround.

System Requirements

80386-based computer such as IBM® PS/2™ Model 80 or COMPAQ® DESKPRO 386® or 80286-based computer with Intel® Inboard™ 386/AT.
One 1.2 Mb (5 1/4 ") or one 3 1/2 " microfloppy disk drive.
One hard disk drive.
DOS 3.0 or later.
2 Mb RAM recommended.
Supports monochrome, CGA, EGA, VGA, and Hercules™ monitors.
Not copy protected.

The Professional MultiTasker™



IGC
4800 Great America Parkway
Santa Clara, CA 95054
(408) 986-8373

VM/386 is a trademark of IGC. IBM is a registered trademark of International Business Machines Corporation. COMPAQ and DESKPRO 386 are registered trademarks of Compaq Computer Corporation. Intel is a registered trademark and Inboard is a trademark of Intel Corporation. Hercules is a trademark of Hercules Computer Technology. AutoCAD is a trademark of Autodesk, Inc.

News & Views

by Sol Libes

Random Rumors & Gossip

IBM is expected to begin shipping Version 3.4 of DOS, possibly as early as this month. It will offer a windows-based shell, extended memory and display drivers, and hard-disk partitions up to 512 Mbytes. Written by IBM, it will be licensed to Microsoft for relicensing to other OEMs.

Intel has begun shipping samples of its 16-bit I/O 80386-compatible processor chip to OEMs with production quantities expected as early as October. The likelihood is that **IBM** and **Compaq Computer** will use the chip in new laptops to be formally introduced next month.

Maxtor expects to begin shipping the first erasable optical disk drive in September. It will be a 5 $\frac{1}{4}$ -inch unit with a 1-gigabyte capacity. A prototype was shown at Comdex in May.

Apple's next new product announcements, expected in the late fall, will likely include an SE+ that supports color and uses a 68020, and a Mac III that runs on a 68030. Also expected are Mac II enhancements that will include new color display and communications controller cards, and a new version of the operating system that supports these cards. Apple's laptop Mac is rumored to be back on the drawing boards after prototypes shown to key customers got thumbs down reviews—look for a possible intro in January.

NeXT Computer systems are reported to have been shipped to beta testers in early May, although the final configuration had not yet been determined. The operating system is expected to be the MACH UNIX environment from Carnegie Mellon. A formal introduction was expected before July 1.

Compaq Computer is estimated to now have a 60 percent share of the 386 system marketplace and IBM only a 10 percent share.

There is speculation that **Intel's** new 80486 microprocessor chip will allow running multiple OS/2 sessions under UNIX as the chip will support 286 virtual mode. There are also rumors that at least one version of the chip will be pin-compatible with the 80386, and can possibly be used as a direct replacement.

Expect **Ashton Tate's** long-awaited introduction of dBASE IV to

be delayed again. Look for the current release date of July 31 to be pushed back into September, or even later.

Hitachi has demonstrated a prototype 6.3-inch laptop color LCD display and is promising to follow that with a 10-inch unit. Samples are expected for year-end and production should begin by the middle of next year.

Micros Dominate the Computer Business

The Data Analysis Group, a market research firm, reports in its "Computer Industry Forecasts" that micros are now the largest segment of the computer business, with annual sales of \$25 billion compared to \$16 billion for mainframes, \$14 billion for minis, and \$2 billion for supercomputers. In terms of installed systems, the report says there are:

- 12,000,000 PC/MS-DOS systems
- 3,000,000 Apple II systems
- 1,500,000 Macintoshes
- 600,000 UNIX systems
- 75,000 IBM System 36s
- 25,000 VAX 785 and larger machines (not workstations)
- 20,000 IBM and compatible mainframes
- 12,000 IBM System 38s
- 1,200 supercomputers and mini-supercomputers

UNIX Spreads

Commodore will soon join the ranks of UNIX systems vendors with the Amiga 2500, a prototype of which has been shown to potential European customers. Expected to ship in September, it will be 68020-based; have 3 Mbytes of RAM; 40-, 60-, or 100-Mbyte hard disk, and sell in the \$5,000 to \$6,000 range (base price). Commodore will also offer a TOPS network linking the Amiga and Sun Microsystems workstations.

IBM is expected to begin shipping its AIX version of UNIX for the PS/2 Model 80 in September. **SCO Xenix/386** for the **Compaq DeskPro 386/30** has been available for some time now. And **Sun Microsystems** began shipping its Sun386i/150 in April. These three systems are expected to dominate the 386-based UNIX workstation marketplace. All will support DOS running as a task under

UNIX. The Sun system appears to be the most aggressively priced at under \$11,000, compared to similarly equipped Compaq and IBM systems selling for \$14,700 and \$17,400.

OS/2 News

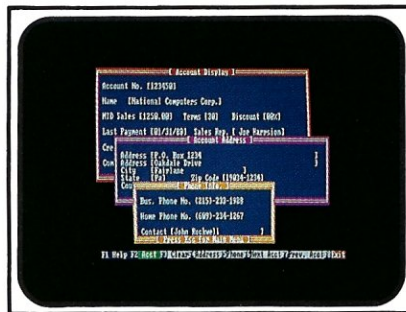
IBM has boasted that it is spending \$200 million a year on OS/2 development and expects to ship Version 1.0 of the OS/2 "Extended Edition" this month, several months earlier than originally promised. The initial version will include the Database and Communications Managers (developed by IBM) but not the Presentation Manager (PM) user interface (being developed by **Microsoft**). The PM is expected to follow in Version 1.1, whose delivery has been pushed back to October. Although IBM has demonstrated the PM, the company and Microsoft are struggling to remove bugs and improve performance. Microsoft is expected to ship its version of the PM to OEMs the same month, which most likely means that companies such as Compaq, Zenith, and AST will not offer the PM until early next year.

OS/2-EE is an IBM-developed extension of Microsoft's OS/2 and is intended primarily to provide connectivity to IBM's larger systems. Several clone producers are already boasting that OS/2-EE will run on their PS/2 compatibles. This speculation is based on the fact that OS/2 does not access any hardware directly, but makes all calls via the operating system. However, there is speculation that the Communication Manager may access a new multi-master network interface directly. If the network interface is on a plug-in card, users may have to purchase the card from IBM. If the network interface is on the motherboard of IBM's new PS/2 systems, the cloners will have their work cut out for them and the interface will take time to develop.

It is interesting to note that Microsoft's version of OS/2 differs from the IBM implementation in that it bypasses most of the ROM BIOS. OEMs who use Microsoft's OS/2 must customize it for their hardware configuration. Some clone manufacturers who have a high degree of IBM hardware compatibility are not electing to offer their own OS/2 implementation and are recommending that users purchase IBM's OS/2. IBM should find that it is selling a lot more copies of OS/2 than originally expected.

IBM is also expected to add "named pipes" to its OS/2 LAN Server, which will make it easier for software developers to write LAN applications

Don't fall flat on your Ashton.



Power up Q-PRO4™ Version 5.0. Get all the reliable, 'bug-free' features you need NOW!

The Most Powerful Screen Handler You've Ever Seen.

Q-PRO 4 offers all the features, the power, and the reliability you need. And it's available now!

- What you see is what you get (WYSIWYG) no coding
- Fill-in-the-blanks screens
- Field editing with attributes
- Entry fields addressable as variables in the program
- Yes/no fields
- Multiple choice fields
- Entry fields symbolically addressable
- Programmable screen field types
- Picture data editing
- 16 color palette
- Active field color
- Painted background

Fully Featured Windows (optional)

- 100 windows per screen
- Multiple windows displayed simultaneously
- Frame and title
- Any size, any location
- Create WYSIWYG with included editor
- 3 Types: data entry, scrolling help, and data display

No Limits

Unlimited files open simultaneously (we have overcome the DOS limits).

- 2,000,000,000 bytes per file
- Unlimited record size
- Unlimited number of fields on screen
- Unlimited memory arrays
- Unlimited size of one and two dimensioned memory arrays

Extraordinary Data Files

- Data dictionary
- Nine indexes per data file
- All indexes in one file
- Ascending and descending indexes
- B+ tree (CBTREE) state-of-the-art speed
- Also random, ASCII, and comma delimited

Advanced Fourth Generation Language

- Event driven architecture
- Procedural, with over 120 commands
- Lightbar menus
- Global subroutines
- Global variables
- Directory and path management
- Two dimensional tables
- Variables: string, numeric, floating point, Boolean
- DOS shell
- On line debugger
- Julian dates
- Sophisticated editor included

Report Generator

- Relational, multifile WYSIWYG

Modern Transaction Processing

- Transaction recovery and rollback

Security

- Optimizer encrypts and speeds up programs
- Data encryption
- Screen security
- Easy, consistent, predictable, corrupted file recovery

LAN and Multiuser

Record and file lock for:

- Novell Network
- IBM PC-Network
- 3COM 3PLUS
- Software Link PC-MOS

Join the next generation! Order your Q-PRO 4 today.

Only \$795.00 with windows.

\$495.00 without. \$50.00 evaluation

Q-n-e™
international

136 Granite Hill Court
Langhorne, PA 19047

(800) 333-0448

TLX 291765

Power Graphics

Essential Graphics Takes You To New Heights Of Graphic Programming In C. Increases Speed 40%.

When first brainstorming this ad I spent a considerable amount of time trying to determine what graphic image to use as an illustration. The Space Shuttle, Mona Lisa, Robo-Cop - there are so many available.

Then it occurred to me. When you have the fastest, smallest functions, it's really irrelevant to show a complicated graphic image. It would be as if thinking up a sexy graphic were the test of a library.

The Graphics Test

The crucial test of a professional graphics package is: are the functions powerful, reliable, fast, and do they truly eliminate grunt work?

How quickly the functions execute is the criterion most people look for in a graphics library. There is no sense paying for a package that is not up to speed.

Beware Of Speed Traps

We eliminate the bios calls and write directly to the graphics card. As a matter of fact, in a recent benchmark, we were clocked 40% faster than our nearest competition.

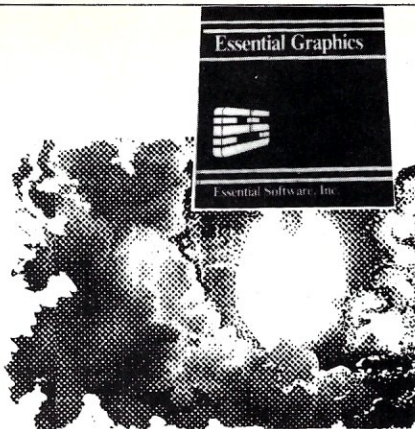
I'd like to repeat that "...clocked 40% faster than our NEAREST competition." Please take a moment to think about the significance of that speed increase in the project you are contemplating or working on now.

Our efficient, granular coding provides you with code sizes up to 75% smaller. Lean, fast and tight - just the way you would have done it yourself.

Power Packed Pixels In Every Package

There has always been a trade-off in this industry between ease of use and power. Our functions do not require a lot of setups, are well-documented, and most of all, thoroughly debugged. Essential Graphics' ease of use stems from our thoughtfulness and not from a lack of power. We explain what we are doing every step of the way. Our support staff consists of the humans who wrote the functions, so we are thoroughly prepared to assist you after the purchase.

Essential Graphics is a trademark of Essential Software



Caveat Emptor

Make no mistake, this is not a package for the "draw a box around the total field" crowd. This library was designed to help the professional C programmer make money and look good.

We've included a complete set of "rubber-banding" functions. One of the most welcome features is the ability to save/restore images in PC Paintbrush format or bit image. World coordinates and view ports aid in programming portability.

We include the ability to manipulate and rotate character fonts and symbols. You can place characters and symbols anywhere on the screen, and use up to eight fonts at one time.

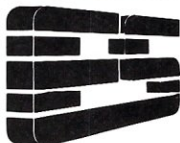
Yours, Mine, Ours

We don't consider ourselves equity partners in your business and therefore we do not charge any royalties or run time fees. We think your efforts belong to you. If for any reason you are unsatisfied with our product you may return it within 30 days for a full refund. Full source is available. Please call today and launch yourself into the world of power graphics.

Price \$299 - Source \$299

Adaptors include - CGA, EGA, VGA, MCGA, ATT, ATT DEB, Hercules, Vega Deluxe, Paradise Autoswitch. **Printer Support** - IBM, Epson, Oki, TI, Alps, Panosonic, and others. **Supports** mice, light pens, plotters, color printers. **Compilers**-Microsoft, Lattice and Turbo-C

Other Essential Products Include: ScreenStar - Essential Communications and Utilities -- /*resident_C*/ - Please call for further information 201-762-6965.



Essential Software, Inc.

South Orange Plaza
76 S. Orange Ave., Suite 3
South Orange, N.J., 07079

**To Order Call:
201-762-6965**

as well as provide software compatibility to Novell and Microsoft LAN operating systems.

Microsoft's Bill Gates has disclosed that OS/2 support for the MicroChannel Architecture's multi-master operation is still in a very early stage of development and that OS/2-386 is slated for release next year.

80386 Compatible Chip Announced

A rumor published in last month's *News and Views* column has been confirmed. **VM Technology** of Tsukuba, Japan, has unveiled an 80386-compatible processor chip called the VM-8600S. It uses a programmable logic array and hence the company feels that it is not infringing on Intel's 80386 microcode patent. The company expects to begin sampling the product in the United States in September and claims the chip's performance is equal to that of the Intel chip.

So far, Intel has refused to license the manufacture of 386 chips to companies other than IBM. 386 chips have thus been in short supply. The availability of this chip is expected to cause a significant decrease in the cost of 386 systems.

Number of Computer Stores Decline

Over the past two years, a substantial number of computer stores have closed, gone bankrupt, or been consolidated with other stores. And, the trend appears to be continuing. In 1987, 663 computer stores reportedly ceased operation while only 114 new stores opened. In 1986, 948 stores went out of business and 359 new stores opened.

This trend is attributed to discounting, and manufacturers' aggressive dealer quotas and revenue incentive programs. The result has been falling prices and thin margins.

Newsletters of Note

The following are some interesting newsletters that *Micro/Systems* readers may consider of interest. Call for a sample issue and for subscription information.

- *Micro Software Marketing* by Scott Whitt; (919) 268-5925
- *Microprocessor Report* by Michael Slater; (415) 494-2677
- *Softletter* by Jeff Tarter (617) 868-0157
- *Software Success* by David Bowen (408) 446-2504
- *The Programmer's Letter* by Bruce Lynch (617) 740-2452 □

CIRCLE 43 ON READER SERVICE CARD

Simply the **BEST C and Pascal** on **AT, 386, Sun, Apollo, RT, VAX, 370**

"The most rock-solid C compiler in the industry. Superb technical support and portability. Superior code generated."

Gordon Eubanks, Symantec — Q&A (386).

"It simply works, with no trouble, no chasing strange bugs, and excellent warning and error messages ... a professional product."

Robert Lerche, Bay Partners.

"For large-scale software development, the highest quality C compiler available on the market today. Pragmas are great. Quality of support is exceptional." **Randy Neilsen, Ansa—Paradox (DOS, OS/2).**

"15% smaller and 15% faster than Lattice C."

Robert Wenig, Autodesk.

"Our software is running anywhere from 30 to 50% faster than when compiled under Lattice." **David Marcus, Micronetics.**

"We switched from Lattice due to a 10% reduction in code size. The compiler is very stable." **Lee Lorenzen, Ventura Software — Ventura Publisher, marketed by Xerox Corp.**

"Best quality emitted code by any compiler I've encountered. Often amazing." **Bill Ferguson, Fox Software — FoxBase (386).**

"Messages sometimes pointed out type mismatches, incorrect-length argument lists, and uninitialized variables that had been undetected for years [in 4.x bsd]." **Larry Breed, IBM ACIS [RT PC].**

"Diagnostics turned up bugs missed by other compilers. Rapid bug fixes by technical support, someone who knew what he was talking about. 80386 code is well optimized."

Tim Addison, Logistics Data Systems.

"386 protected mode support is fantastic, especially the access to large amounts of memory. It's mainframe compute power on a PC."

Dan Eggleston, Viewlogic.

"The preprocessor supplied with Professional Pascal is quite useful. The code quality and control over segmentation and memory models are superior to MS Pascal." **Bob Wallace, QuickSoft.**

Check Out These Reviews

• High C™:

<i>Computer Language</i>	February 1986, '87	
<i>Dr. Dobb's Journal</i>	August 1986	
<i>PC Magazine</i>	Jan. 27, 1987	(80386 version)
<i>Dr. Dobb's Journal</i>	July 1987	(80386 version)
<i>BYTE Magazine</i>	November 1987	(80386 version)

• Professional Pascal™:

<i>PC Magazine</i>	Dec. 29, 1985
<i>Computer Language</i>	May 1986
<i>PC Tech Journal</i>	July 1986
<i>Journal of Pascal, Ada, & Modula-2</i>	Nov.-Dec. 1986
<i>BYTE Magazine</i>	Dec. '86, June '87 (80386 version)

Why MetaWare compilers

- They are specifically designed for serious software developers.
- They are reliable and robust: they don't break at every turn.
- Their generated code is the best, or near best, on each architecture.
- Their superior diagnostic messages help you produce better products more quickly.
- Your source can be ported with ease to the most popular systems.
- You can link mixed-language modules from our compilers, others.
- You can benefit from high-level, personal technical support.
- You can take advantage of the latest ANSI C extensions, and/or extensive Pascal extensions. **High C** has been tracking the ANSI Standard for two years; **Professional Pascal** will soon have a VS Pascal compatibility switch and several Apollo Pascal ext'ns.
- You can take advantage of the latest 387 and Weitek 1167 support — we have the only compilers with Weitek real mode support.



Power Tools for Power Users

Ashton-Tate: dBase III Plus, MultiMate; Autodesk: AUTOCAD, AUTOSKETCH (8087, '387, Weitek); Boeing Computer Services (Sun); CASE Technology (Sun); CAD/CAM giant Daisy ('86, '386, VAX); Deloitte Haskins & Sells; Digital Research: FlexOS; GE; IBM: 4.3/RT, 4680 OS; Lifetree Software (Pascal); Volkswriter Deluxe, GEM-Write; Lugaru: Epsilon; NYU: Ada-Ed cmplr; Semantec: Q&A; Sky Computers; ... (Product names are trademarks of the companies indicated.)

Industrial-Strength

MetaWare C and Pascal compilers are designed for professional software developers. These tools are loaded with options to control them for special purposes. You can adjust the space-time trade-off in code quality. You can adjust external naming conventions to agree with linkers and operating systems. You can specify segment names for segmented architectures, and to help place code or data in particular places for embedded applications. You can select from five memory models for the 8086 family. And on and on.

A Partial List of Optimizations

Common subexpression and dead-code elimination, retention and reuse of register contents, jump-instruction size minimization, tail merging (cross jumping), constant folding, short-circuit evaluation of Boolean expressions, strength reductions, fast procedure calls, automatic mapping of variables to registers (where advantageous), ...

"Platform" — Code Quality

Sun, Apollo, SGI — 18%, 3%, 26% > resident compiler (Dhrystone).
 PC: DOS, OS/2 — 3-10% > Microsoft C; 30% > MS Pascal, LatticeC.
 386 32-bit DOS — no competitors, since November, 1986.
 286, 386 UNIX — 66% better than pcc (Dhrystone, 386).
 VAX VMS — ≈ DEC's excellent C and Pascal; better features.
 VAX Ultrix — 19% > pcc (Dhrystone); much > Berkeley Pascal.
 RT PC/4.3bsd — 89% > the original port of pcc (Dhrystone).
 370 CMS, UNIX — much better than any C, and VS Pascal.
 AMD 29000 — >40,000 Dhrystones! Available in Q2, cross.

(408) 429-6382, telex 493-0879.

Since 1979.



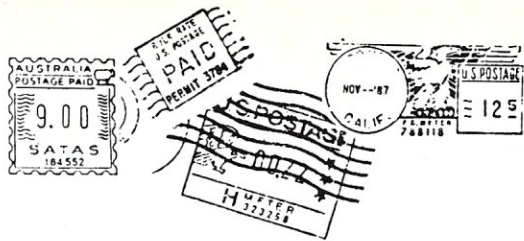
903 Pacific Avenue, Suite 201 • Santa Cruz, CA 95060-4429

The Clear Choice for Large Programming Projects

— PC Tech J.

© 1987 MetaWare Incorporated. MetaWare, High C, Professional Pascal, and DOS Helper are trademarks of MetaWare Incorporated. Others and their owners are duly respected.

there is mail . . .



We welcome your letters with any comments about articles you have read or would like to read in *Micro/Systems*. Please address your comments to *Micro/Systems Journal*, 501 Galveston Drive, Redwood City, CA 94063. Letters to the editor will not be published if they do not include a name and address. All letters become the property of *Micro/Systems Journal*.

RTOS Update

Dear *Micro/Systems*:

I would like to clarify a couple of items from Stuart Jones' excellent article on *MicroWay's* RTOS (*Micro/Systems*, March 1988). In both cases, I believe the errors simply represent the time lag between writing and publishing the article.

First, the current version of RMX-86 is 7, not 6, with Release 8 due out shortly. More importantly, RMX-286 has been available for more than a year and is currently at Release 2, Update 3. We use RMX-286 as an operating systems teaching vehicle at Queens College, and our students have been working with this protected mode OS for three semesters now, doing projects that range from extensions of the Command Line Interpreter to interrupt-driven device drivers.

The second point is a minor technical matter. Round robin task scheduling with a user-configured time slice has been an option for

some time now. Task switching is not necessarily "entirely event driven" as distributed by Intel.

We have found that RMX provides a rich yet well-structured environment for teaching current OS concepts. Dr. Jones' article does a fine job of distilling an introduction to RMX into five or six pages.

Christopher Vickery
Associate Professor
Queens College
Flushing, N.Y.

Lost in Novell Gnet? You CAN Go Home Again!

Dear Editor:

I have observed that there are features on some operating systems that endear themselves to users very quickly. Commands such as Alias, Set, Unset, and Home can make life for the user very enjoyable. Life on other operating systems may not be so pleasant, but with a little creative thinking, the situation can be improved.

As a lecturer in Software Engineering, I always like to throw examples to my students to illustrate how to make systems more user friendly. Here is one of those examples that I trust you will find useful.

In our network, each student is assigned a unique user name and home directory where he or she has full rights to create subdirectories, delete files, etc. Being what they are,

students tend to be curious and spend many hours browsing through the system.

Upon occasion, problems may arise and a user could find himself (or herself) left in some foreign directory. To help these lost users find their way home, we decided to write a small program to help them get back to their own directory. Listing 1 provides a program that searches the environment table for the variable *HOME*, and then initiates a change drive/directory command via DOS to relocate the user to the original login directory. This utility has been extensively tested on our network running PC-DOS Version 3.2 and Novell Gnet Version 2.0a on an AT-compatible file server.

In order to use this utility, however, the system supervisor must log in each unique user to his or her own directory, and include the appropriate commands in the user's login script using the *syscon* utility (or *setlogin* if you have an older version). Listing 2 shows a suitable login script for the unique user BRIANB.

Using this same technique, we have created other utilities. One is called *local* to transfer similar users to a group directory that contains course-specific details such as assignments.

Brian Brown
Central Institute of Technology
Wellington, New Zealand

How Do You Use IND\$FILE?

Dear Editor:

In Michael Cherry's *LANscape* column "3270 Gateways—Connecting IBM Mainframes to PCs" (*Micro/Systems*, March 1988) he talked about using *IND\$FILE* to transfer files. I have been trying to find information concerning *IND\$FILE*, but no one, including IBM, will tell

... continued on page 67

Listing 1. HOME.C: A "Go Home" Network Utility

```
/* HOME.C (c) B Brown, 1988, Compiled using TurboC v1.0*/

#include <dir.h>
#include <stdio.h>
#include <stdlib.h>
#include <alloc.h>

main()
{
    int drive;
    char *homedir;
    if (9 & homedir = malloc(20)) == NULL)
    {
        printf("\nhome: Error allocating memory.\n");
        exit(1)
    }

    homedir = getenv("HOME");
    drive = homedir[0] - 'A':
```

```
setdisk( drive );
if( chdir( homedir+2 ) == -1 )
{
    printf("\nhome: Error restoring home directory.\n");
    exit(2);
}
free( homedir );
}
```

Listing 2. A login Script for Unique Network Users

```
REM Login Script for user Brian Brown
MAP DISPLAY OFF
MAP SEARCH8:=SYS:U0\LOCAL
MAP D:=SYS:U0\BRIANB
SET HOME="D:\U0\BRIANB"
MAP SEARCH9:=D:
MAP DISPLAY ON
DOS BREAK ON
DRIVE D:
```


Concurrent DOS 386

Think small in a big way

When you think multiuser/multitasking, think Concurrent™ DOS 386, the big name in small systems from Digital Research, architects of the first standard operating system for personal computers. Now, Concurrent DOS 386 allows multiple users to share peripherals, files and applications, using serial terminal workstations linked by RS-232 cables to the system. It's fast, reliable and economical.

The big news today is small systems

Concurrent DOS 386 meets the increasing demands placed on small systems by supporting multiple DOS programs on both the system console and attached terminals. You can run popular programs such as Lotus® 1-2-3®, dBase® III, WordPerfect® and many more, with full math coprocessor support. The system runs up to 255 tasks simultaneously, with full intertask communications and byte-level record, file and device locking.

For people who hate waiting in line

Concurrent DOS 386 brings you all the remarkable speed and power of the Intel® 80386 processor. A prioritized pre-emptive scheduler allows task execution

and intertask communication by several users at near full processor speed while letting some tasks "interrupt" others according to the needs of each user.

A small system with a big memory

Concurrent DOS 386 gives you access to four gigabytes of linear physical memory. Its powerful memory paging capability fully supports the Expanded Memory Specification with no additional hardware or software.

Menus at a touch

Now you can create and customize menus, while programmable function keys let you condense complex commands to a single keystroke. The file manager runs standard operating system functions, plus you have an on-line help facility, text editor and support for DOS-based device drivers.

Multiuser color graphics

Now with the introduction of the newest member of the Concurrent DOS family, Concurrent DOS 386/Multiuser Graphics Edition, your demands for high-resolution EGA bit-mapped graphics in the workstation environment can

be met. Take advantage of advanced technology allowing you to run popular DOS-based graphics programs on individual workstations as well as on the system console without sacrificing system performance. Ask us about this exciting new version of Concurrent DOS 386.

All you have to remember is Concurrent DOS 386

Concurrent DOS 386 from Digital Research is the name to remember when it comes to 386 technology. The power and versatility of Concurrent DOS 386 are giving a new meaning to the word multiuser.

**CALL DIGITAL RESEARCH AT
1-800-443-4200 AND ASK FOR OUR
CONCURRENT DOS PROGRAMMER
INFORMATION KIT.**

**CONCURRENT DOS 386:
SHARING THE SYSTEM AFFORDABLY**

Digital Research and the Digital Research logo are registered trademarks, and Concurrent is a trademark of Digital Research Inc. Other product names are registered trademarks or trademarks of their respective owners. Specifications are subject to change without notice. Copyright © 1988, Digital Research Inc. All rights reserved.

 **DIGITAL RESEARCH®**

CIRCLE 98 ON READER SERVICE CARD

by Don Libes

Something for Everyone

Rather than presenting one particular programming subject in detail as I normally do in this column, I am going to present several tips from those I constantly collect. They aren't appropriate for a whole column, but I do want to share them with you. I'll be surprised if you don't use at least two of them in your next program.

Sometimes I find neat ideas in my colleagues' programs. Often I get good things from readers. Please don't send me any massive programs but if you have any good tips, send them to me and I'll write about them next time!

1) Roger Hayes of the Department of Computer Science at the University of Arizona makes the following suggestion. Write conditional expressions with the constant first, as in:

```
if (10 == j)
```

rather than

```
if (j == 10)
```

If you ever miscode `==` as `=`, the compiler will complain about the first form, but not the second. You may have to think about it the first couple of times you write it, but it will quickly become automatic.

2) K.W. Heuer of AT&T Bell Laboratories, Liberty Corner, New Jersey, improves on the old trick of defining *streq* as a macro. He suggests that *streq* can be generalized as follows:

```
#define strrel(a,R,b)
  (strcmp(a,b) R 0)
```

Now you can write code like

```
if (strrel(a,==,b))
```

or

```
if (strrel(a,<,b))
```

It looks kind of weird at first, but its meaning is clear. Some might consider this to be an abuse of the

preprocessor, but I think it is very clever!

3) Many readers continue to rediscover the following trick. Suppose your program uses a large static array, such as:

```
static char memory[BIG_NUMBER];
```

Some compiler/linker combinations actually allocate this space on the disk in the executable program, but there is no need to do that. Instead, it can be set up at run time when the program is loaded:

```
static char *bigz;
/* add the following to main*/
char *bigz;
bigz = malloc(BIG_NUMBER);
for (bigz=bigz+BIG_NUMBER1;
     bigz>=bigz;bigz--)
    bigz = 0;
```

Here we allocate and zero the space during program initialization. Space will be saved on disk, and time will be saved during program loading. If you don't depend on the space to be zero, you can remove all the references to *bigz*.

Consider using this tactic with large arrays even if they are initialized. Initialized arrays are guaranteed to take up space on disk, while it may be easy to replace them with a couple lines of executable code.

4) Guido van Rossum of Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands, suggests the following function to quickly count the number of 1's in a bit string (assume a 32-bit value for simplicity):

```
return count[x&0xff] +
       count[(x>>8)&0xff] +
       count[(x>>16)&0xff] +
       count[(x>>24)&0xff];
```

Count is a 256-byte array of pre-computed values. Store it in an object library so that you can easily use the array without redefining it each time. Here is the beginning of the declaration (hopefully, it should be obvious how to continue it):

```
char count[256] =
{0,1,1,2,1,2,2,3,1,
```

5) How do you take *sizeof* of a *struct* member without an instance of it? For example, suppose you have the following declaration:

```
struct s {
    struct a {
        ...
    };
};
```

If you want to get the size of *a*, you need to have an instance of *s*. You can't say *sizeof(struct s.a)*, or something along those lines. The only other alternative is to extract the whole declaration of *a*. But you want to avoid unused variables or declarations, as well as decrease the possibility of rewriting it incorrectly the second time. (And *lint* will complain, too.)

Wayne Throop of Data General suggests the following construct:

```
sizeof(( (struct s *)0 )->a)
```

This solves the problem very nicely. It looks strange because it appears to be dereferencing a null pointer. However, *sizeof* is defined not to produce side effects, so it is perfectly legal.

6) Ron Vaughn of AT&T Bell Laboratories, Indian Hill, Illinois, suggests a "dirty workaround" to enable passing *arrays* by value. The problem is that *arrays* in C cannot be passed by value; an *array* name just stands for its address, and putting an asterisk in front of it just returns its first element. However, *structs* can be passed by value. The solution is to embed the array in a *struct*:

```
struct {
    char array[100];
} dummy;
```

It seems kind of degenerate since the *struct* only has one element in it, but the fault is in the language, not the solution. This is indeed what everyone does to pass *arrays* by value.

7) How many bits are in an *int*? Surprisingly, this question comes up a lot. For example, you need to know it if you want to extract the most significant bit. It is a mistake to use the expression *(sizeof int)*8*, because not all machines have 8-bit bytes. (Strictly speaking, 8-bit bytes are called *octets*, but unqualified bytes may be any length.)

ANSI X3J11 defines a symbol called *CHAR_BIT* that defines the number of bits in a character. But this is not necessarily the same number as are in a byte. A routine supplied by Dave Luke of The Instruction Set Ltd., London, England, does the trick.


```

unsigned int i = ~0;
int nbits = 0;

do
    nbits++;
while((i >>= 1) != 0);

return nbits;

```

8) The last goodie is buried deep (page 223) in *C: A Reference Manual* by Samuel Harbison & Guy Steele (the *other* C book always on my desk). Unfortunately, too many people take the title literally and miss getting everything it has to offer. Under the guise of illustrating basic C concepts, the authors include several gems.

One of them is to find the smallest element in an array "a" of length "n." The obvious approach is:

```

smallest = a[0];
for (i=1; i<n; i++) {
    if (a[i] < smallest)
        smallest = a[i];
}

```

Harbison & Steele point out that each iteration of the loop includes a test for the termination condition, plus a test to see if a new smallest value has been found. Their clever

rewrite obviates the test for a termination condition (most of the time):

```

int temp_a0 = a[0];
int smallest = a[0];
int *p = &a[n];

for (;) {
    while (*--p > smallest);
    if (p == &a[0]) break;
    a[0] = smallest = *p;
}
a[0] = temp_a0;

```

The basic idea is to scan down the list looking for ever smaller values without checking for the array bound. The reason we don't have to worry about falling off the end of the array is that the end of the array always contains the currently known smallest value.

Any time we get a new smallest value, we move that to the end of the array. The final line restores the array to its original value, and *smallest* holds the smallest value.

9) As long as I'm talking about books, I'll mention that the C programmer's bible, *The C Programming Language* by Brian Kernighan and Dennis Ritchie (K&R), has just been released as a second edition. The new edition is a complete rewrite

of the original K&R (also known as the "white book"). It is based on the ANSI X3J11 standard and includes new examples, a simple version of *cdecl* (see *Micro/Systems Journal*, January/February 1987), discussions of the C language changes, and a *yacc*-able grammar. And the cover is still white. This is *the* book to learn C from. It is destined to become a classic.

The first edition will remain in print, as it differs significantly from the second and describes many of the extant C implementations. You might be amused to learn that the original expectation of Prentice-Hall was that an average of nine copies of K&R would be sold to each of the 130 UNIX sites that existed at the time the book was first published. At last count, over a million copies of the first edition had been printed!

Harbison & Steele and both editions of K&R are available from Prentice-Hall, Inc. □

Don Libes is a computer scientist working on artificial intelligence in robot control systems in the Washington, D.C., area.

Did you find this article particularly useful? Circle number 1 on the reader service card.

The Custom 386 Programmer's Workstation

Looking for a lightning-quick 386 system that's tailored to your needs? CAE/SAR Systems, Inc. will custom-fit you a 386 system more powerful than most on the market. Whether it's a system designed for your program development, artificial intelligence, CAE, or systems design work, CAE/SAR delivers reliable, powerful 386 workstations built for today's programmers.

Based on a proven 386 motherboard, CAE/SAR 386 systems come in dozens of different configurations for memory, disks, floating point and graphics. You can select high speed drives (16 ms), 70Mb, 140Mb, or 300 Mb; EGA or mono monitors and cards; and 2.5 Mb, 4.5Mb, or 8.5Mb 32-bit RAM—plus other options!

The CAE/SAR 386 systems run Unix and DOS concurrently, and also run OS/2

and Xenix. Floating point options are available for the Intel 387 chip.

Basic Unix/Xenix systems start at \$3,495.

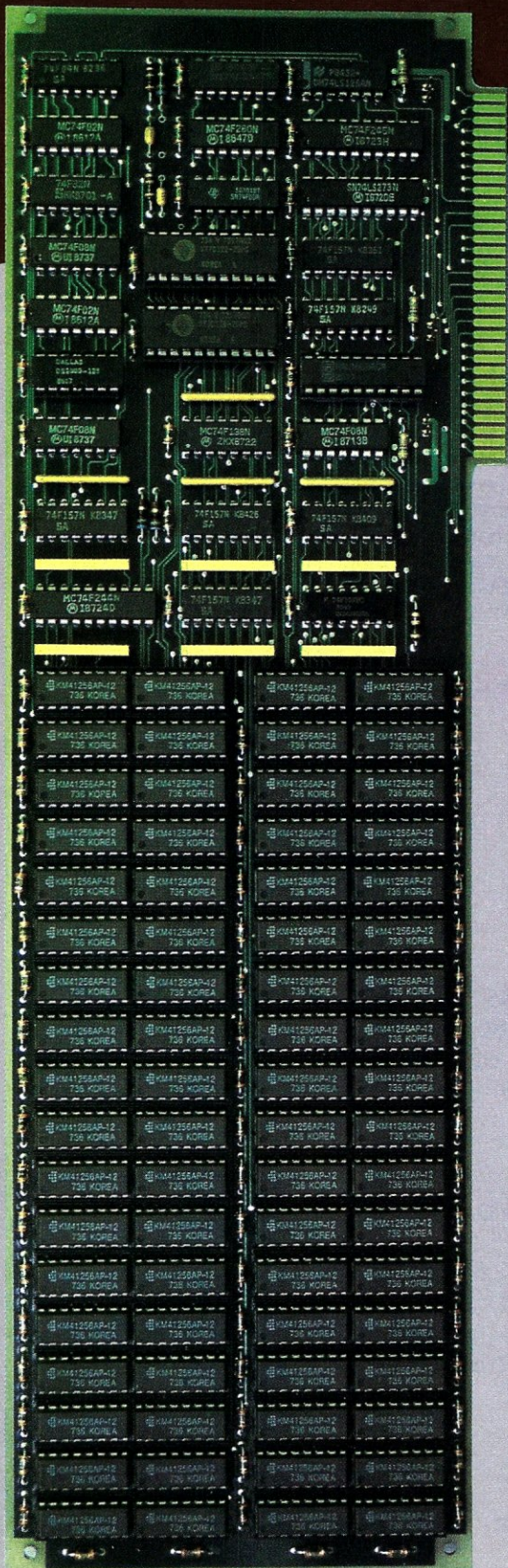
Get a system that fits you perfectly. Call CAE/SAR Systems today for more information.

CAE/SAR Systems, Inc.
P.O. Box 50243
Palo Alto, CA 94303
(415) 949-3816

CIRCLE 58 ON READER SERVICE CARD

X-BANDIT

Designed for the EMS 4.0 Standard



DESIGN PHILOSOPHY

- The Teletek X-Bandit was specifically designed to utilize the advanced features of the Lotus/Intel/Microsoft EMS 4.0 Specification. It is available in both 8 and 16 bit versions for use in the IBM XT, AT, and compatibles.

MEMORY

- Segmented Memory Mapping allows the user to fill out unused memory segments between 640K and 1 Megabyte.
- Split Memory Addressing allows the user to fill out conventional memory to 640K.
- Extended Memory Addressing is available for the PC/AT version.
- 2 MB capacity in a single slot. Up to 8 MB per system.
- Parity checking.

SOFTWARE

- Easy menu-driven auto configuration software.
- Device driver includes print spooler and RAM drive.
- Supports multitasking with the appropriate shell-resident software package.

SPEED

- 6/8/10/12 MHz speed with 0 wait states. 16 MHz speed with 1 wait state.

WARRANTY

- One year parts and labor.

TELETEK

4600 Pell Drive
Sacramento, CA 95838
(916) 920-4600
Telex #499-1834

by Patrick H. Corrigan

Networking via Telephone Wire

For several years, I have been telling LAN resellers and installers that if they stay in the networking business, they will also have to get into the telephone business. What I was really saying was that phone systems and LANs would be using the same cabling systems. This prediction is quickly coming true.

Telephone systems primarily use unshielded twisted-pair wire for local connections, while fiber optics are becoming the medium of choice for long lines. More and more, LANs are providing installers with the option of using unshielded twisted-pair cable for departmental links and fiber optics for the network "backbone." In addition, with the decreasing cost of digital data telephone links and the increasing functionality of remote LAN bridges and gateways, LANs and telephone systems are becoming interconnected.

Using standard phone wire (as opposed to shielded twisted-pair or coaxial cable) for high-speed LANs provides several benefits:

1. Greater flexibility—Installers can use the cable that is most appropriate for a given situation.
2. Potentially lower cost—Telephone wire is less expensive to purchase than other cabling alternatives, and usually less expensive to install. Because many twisted-pair LAN components are currently more expensive than their coax-compatible equivalents, costs for LANs based on telephone wire can sometimes be higher. However, with increased usage and production, these costs are dropping.

3. The ability to use existing cable—Because the cost of phone wire is so low, most telephone cable installers run extra phone wire pairs that often can be used for LAN systems.

Phone wire can currently be used with StarLAN, 802.3 Ethernet, 802.5 Token-ring, ARCNET, and others. Let's review some of the available options.

StarLAN

StarLAN is based on a subset of the 802.3 Ethernet standard and utilizes the Ethernet CSMA (Carrier Sense Multiple Access) cable-access protocol. StarLAN is a 1 megabit-per-second LAN designed to run on two standard telephone wire pairs in a hub-and-star or daisy-chain configuration. Although many think StarLAN is strictly an AT&T product, StarLAN components are being made by many companies, including Western Digital, Retix, and Micom/InterLAN.

A fundamental advantage of StarLAN is cabling flexibility. There are three primary ways to cable StarLAN:

1. Star-wired—StarLAN cards may be attached to StarLAN hubs.
2. Daisy-chained—StarLAN cards may be daisy-chained.
3. Combined stars and daisy-chains—Stars and daisy-chains may be used in combination.

To connect StarLAN in a daisy-chain, the "IN" port of one board is connected to the "OUT" port of the next. (Note: Some StarLAN boards have only an "OUT" port. These can be attached to a StarLAN hub or used at the end of a daisy-chain.)

To wire StarLAN as a star, a StarLAN hub is required. This usually has one "OUT" port and 10 "IN" ports. A StarLAN hub functions as a

repeater, with each hub extending the maximum cabling distance by an additional 800 feet. StarLAN devices (hubs and cards) can be chained together in configurations up to 11 levels deep. Each "active" device (a hub or a powered StarLAN board) acts as a repeater, adding another 800 feet of cabling distance to its path. (For a more detailed look, see "The StarLAN Local Area Network," *Micro/Systems*, January 1988).

ARCNET

ARCNET was developed in the 1970s by Datapoint Corporation and is currently being produced by at least 40 manufacturers worldwide. It has an installed base estimated at 750,000 nodes. ARCNET is a 2.5 megabit-per-second LAN that uses a Token-passing access scheme.

ARCNET is usually cabled in a star-cluster configuration, using hubs as distribution devices. ARCNET uses two types of hubs: active and passive. Like StarLAN hubs, active hubs function as repeaters. Passive hubs are only distribution devices and require a reduced cable length. By effectively employing active hubs, an ARCNET configuration can grow very large, up to a diameter of 20,000 feet. Standard ARCNET boards must be connected directly to a port on a hub, and may not be daisy-chained. A relatively recent development is the high-impedance ARCNET card. As many as eight of these boards can be attached to a single ARCNET cable, which then can be attached to an active hub.

Although ARCNET was originally designed to use RG-62 coaxial cable, it can also effectively use twisted-pair telephone cable. Standard Microsystems Corporation (SMC), the developer of the original ARCNET chip set, has recently released a twisted-pair version of its own ARCNET card. Like the high-impedance coax board, this board allows up to eight stations to be daisy-chained using two telephone wire pairs. These chains then can be attached to a special twisted-pair active hub. (At the time of this writing, SMC had not begun shipping the hub.)

Another method of using ARCNET on twisted pair calls for standard ARCNET boards and active hubs. Devices called baluns (BALANCED load to UNbalanced load) are attached to the coax connectors on ARCNET boards and hubs, and then connected to the twisted-pair cable. Nevada Western is one manufacturer whose baluns have been used successfully with ARCNET, and I'm sure there are others. Cable runs can

extend approximately 300 feet between stations and active hubs, or between active hubs and active hubs. Because we are using standard ARCNET components, we can use coaxial cable for greater distances when required.

802.3 Ethernet

Ethernet is a 10 megabit-per-second LAN that primarily uses coaxial cable in a linear bus configuration (see "Extending Ethernet," *Micro/Systems*, January 1988). Several approaches to putting Ethernet on twisted pair have recently emerged. The system that has generated the most excitement is LattisNet from SynOptics Communications in Mountain View, California. SynOptics' networking approach has been adopted almost in its entirety by the 10baseT Study Group of the IEEE 802.3 Standards Committee, the group responsible for establishing a standard for twisted-pair Ethernet.

LattisNet employs a concentrator that is usually attached to a coaxial or fiber-optic Ethernet LAN backbone (although the concentrator can be used without attaching it to a backbone cable). A special LattisNet UTP (Unshielded Twisted Pair) transceiver is attached to the Ethernet

board or device, and then connected to the concentrator using up to 360 feet of dual-twisted-pair telephone wire.

Currently, the concentrators and transceivers can add considerably to the cost of an installation. Two factors may reduce this cost, however. As usage, production, and

*Although
ARCNET was
designed to use
RG-62, it can
use twisted-pair
cable.*

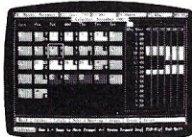
competition increase, the cost of the concentrators should go down. Additionally, SynOptics is cooperating with several Ethernet board manufacturers, including Micom/InterLAN and Kinetics, to incorporate the twisted-pair transceiver into their Ethernet cards.

802.5 Token Ring

Token-ring networks that adhere to the IEEE 802.5 standard, including the IBM Token-ring Network, are designed to use twisted-pair cable, but primarily the shielded, data-grade variety. These systems will run on telephone wire, but only within certain limitations. With cable conforming to the IBM Type 3 specification (which includes most telephone company cable installed over the past few years) the IBM Token-ring will support up to 72 stations, with each station no more than 150 feet from a Multistation Access Unit (MAU), the Token-ring "hub" or cable distribution center). New approaches are appearing, however, that are claimed to break through some of these limitations.

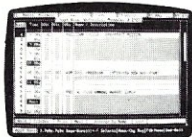
A new Token-ring Multi-Access Unit from Local Data, Inc., of Torrance, California, is specifically designed for use with telephone wire. Instead of the bulky IBM cabling system connectors, these units have RJ-11 telephone jacks that accept standard modular telephone plugs. Depending on the model of Local Data MAU and cable gauge used, Local Data claims cabling distances of 250 to 1000 feet between MAUs and stations.

The Most Powerful LAN Scheduling System You Can Own!



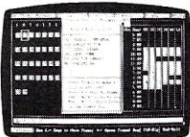
SHARED CALENDAR

Take the entire month at a glance, then explode to the day of your choice - blocked out by hour and minute.



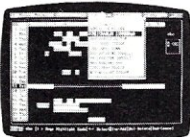
APPOINTMENT BOOK

See when you're booked and when you're free - for a single day or an entire month. Pull-down and pop-up menus guide you. Help is just a keystroke away.



MAKE A MEETING

Schedule yourself and everyone else - simultaneously! Instantly update each participant's personal schedule.



ANALYZE CONFLICTS

See where any conflicts exist - and what you can do about them. Preview schedules by person(s), location, and subject matter.



MULTIPLE PRIVACY LEVELS

Your business schedule remains *your* business. With both password and hierarchical protection, they'll know you're busy but they won't know why.



PRINT IT OUT

Both calendar and appointment list can be printed by individual, location, or subject on a standard dot matrix or laser printer.

It's the on-line, interactive scheduling system for any Novell or MSNET based LAN system. In use by Thousands. There's nothing else like it!

"...a powerful calendar/appointment book that is capable of handling appointment scheduling for a small business or a department of a larger corporation."

- Jon Pepper, PC WEEK

SUM:TIME by Sumware

© 1987 Sumware Inc.

Please send me SUM:TIME. Enclosed is:

- ☐ \$119.95 Single User Version
☐ \$395.00 File Server Version (no limit to stations)

Name _____

Company _____

Address _____

City _____ State _____ Zip _____

Mail to: Sumware, Inc., 23121 Verdugo Drive #101, Laguna Hills, CA 92653

Phone: (714) 855-3062

Designed for the IBM® PC, PC-AT and DOS compatible workstations.

CIRCLE 73 ON READER SERVICE CARD

Other products to extend the capabilities of LANs on telephone wire are no doubt on the way. So if you plan to stay in the mainstream of LAN technology, you, too, may have to get into the telephone business. The cost-effectiveness and increased flexibility offered by plain telephone cable open up a range of new solutions to networking problems.

Next month we will explore this subject even further with a look into the integrator's very own anxiety closet—the phone closet—and examine how we will cable twisted-pair LANs.

Patrick H. Corrigan is a partner in the Corrigan Group—Information Services, an independent consulting firm based in Corte Madera, California, that specializes in local area networking, communications, and office automation.

Did you find this article particularly useful?
Circle number 2 on the reader service card.

Product Information

Kinetics, Inc.
2540 Camino Diablo
Walnut Creek, CA 94596
(415) 947-0998

Local Data, Inc.
2771 Plaza Del Amo
Torrance, CA 90503
(213) 320-7126

Micom-InterLAN
155 Swanson Road
Boxborough, MA 01719
(617) 263-9929

Retix
2644 30th Street
Santa Monica, CA 90405-3009
(213) 399-2200

Standard Microsystems Corporation
35 Marcus Boulevard
Hauppauge, NY 11788
(516) 273-3100

SynOptics Communications
329 N. Bernardo Avenue
Mountain View, CA 94043-5223
(415) 960-1100

Western Digital Corporation
Communications & Subsystems Division
2445 McCabe Way
Irvine, CA 92714
(714) 863-0102

EXPERIENCE THE *INTERACTIVE* APPLICATION DEVELOPMENT SYSTEM

Spend five minutes with the sample applications included, and you will see why more and more developers are now choosing *The Andsor Collection*: attractive, small, fast, fully customized applications, with one tenth the effort.

"With The Andsor Collection we have achieved faster development and more efficient applications, which is important in large and complex projects like our Court Management System."

*Dr. Mark Schrager, Consultant,
Municipal Computer Services, Rochester, New York*

The Andsor Collection: the superb, unified, interactive environment, specifically designed to expedite application development. Ideal for VARs, programmers, consultants.

Fully Featured System

- The convenience of an interpreter, with the speed of compiled applications: among the fastest in the industry
- Many built-in operations, but also the flexibility for complex, custom applications
- Mature, solid, problem-free software: over two years on the market
- Royalty free run-time system
- Easy to learn: use all features interactively before building applications
- Comprehensive documentation: 400 page manual, many examples, sample applications on disk

Self-contained Environment

- Replaces compilers, debuggers, editors, libraries
- One step development: no conversions or translations
- Fully interactive: modify procedures, screens, options, definitions, even while the application is running
- One module, no overlays: small size and fast operation
- The entire application is efficiently stored in one DOS file

Versatile Window Management Functions

- Create tiled, overlapping, stacked, pop-up windows
- Change window position, colors, frame, at any time
- Use windows for data entry, inquiries, help, file maintenance, menus
- Scroll files and screens horizontally and vertically in the window

Powerful Database Management Functions

- Variable length fields and records: simplifies development and saves space
- Use any number of data and index files
- Modify file definitions at any time
- Maintenance-free: no sorts, no reorganization, all files and indexes are updated automatically
- Sophisticated reporting and inquiry capabilities

Beyond File Relations

- Dynamic, open-ended, unlimited relations: multi-file, hierarchical, one to many, many to one, relate a file to itself
- Create different relations between the same files at the same time
- Relations are based on any conditions, not just equal fields
- Use relations in calculations, updates, reports, inquiries, etc.
- No formal definitions: relations are created automatically as files are used together

Flexible Procedural Language

- Use procedures to implement complex applications
- Procedure chaining and nesting, blocks, conditions, loops
- Computational power: expressions, countless built-in functions, data analysis, statistics, date arithmetic, string handling, and more
- Automatic and custom error trapping, recovery, and messages

The Andsor Collection™

buy now and save \$150!

after June 30, 1988

60 DAY
MONEY BACK
GUARANTEE

\$145

\$295

NOW IN ITS
THIRD YEAR

Visa, MC, AmEx, Check

enthusiastic buyers tell us:
it's undervalued!

ANDSOR®

ANDSOR RESEARCH INC.
390 Bay Street, Suite 2000
Toronto, Ontario M5H 2Y2
(416) 245-8073

To order call toll free
(U.S. and Canada)

1-800-628-2828
Ext. 535

Price includes shipping in the U.S. and Canada. Please add \$10 for shipping to other countries. If you return the software, \$8 will be deducted from the refund, to cover our shipping cost.

System requirements: any IBM PC or PS/2 or fully compatible, 320K RAM, one disk drive or hard disk, monochrome or color monitor, DOS 2.0+ or OS/2

© 1988 Andsor Research Inc. Andsor is a registered trademark and The Andsor Collection is a trademark of Andsor Research Inc. IBM is a registered trademark and IBM PC, PS/2, OS/2 are trademarks of IBM Corporation

Assembly Language Primitives for EGA Graphics Under Turbo C

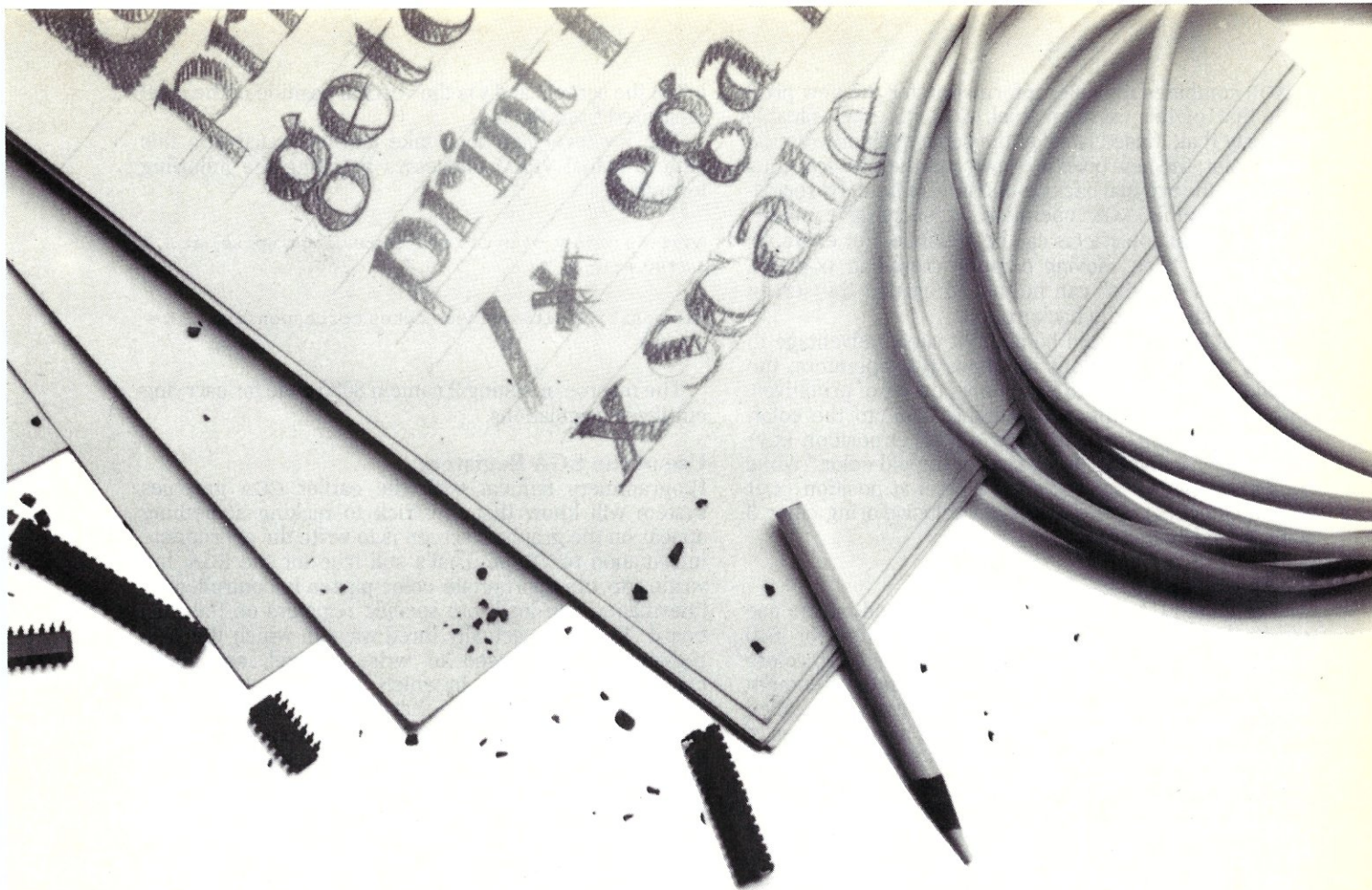
by Thomas Dwyer, Arthur Huang, and John King

The graphic "primitives" presented here are speed-critical, low-level routines written for the EGA that can act as the cornerstone for an entire C graphics library.

Both the original Enhanced Graphics Adapter (EGA) and the recent improvements (e.g., EGA Plus cards and VGA) make it possible for programmers to improve output with color text and color dot-addressable graphics. A good discussion of the various EGA text modes was given by Charles Petzold in *PC Magazine* (August 1986). The second part of his article (*PC Magazine*, September 1986) gave a brief introduction to the use of the EGA dot-addressable graphics capabilities, and offered an example written in C that showed how to draw solid boxes as sets of horizontal lines.

A more extensive set of routines for using EGA graphics under Turbo C was given in *Dr. Dobbs' Journal of Software Tools* (November 1987). The author, Kent Porter, implemented the majority of these routines through calls to the video BIOS services (interrupt 10h), again writing all the code in C. The advantages to this approach are simplicity and portability. The disadvantage is a loss of speed. As we will show, the loss in speed for routines that depend on writing large numbers of pixels to the screen is on the order of one magnitude, that is, speed can be reduced by a factor of more than 10.

Our purpose is to lay the foundation for constructing a general purpose graphics library (much of which also can be written in C), in which the most speed-critical, low-level routines—the so-called graphic primitives—are written in assembly language. Since there are several approaches one can take to programming the EGA at the assembly language level, a simple benchmark for comparing and selecting the best approach was first developed. Five approaches to using assembly language to create a program for writing dots were then compared using this benchmark. The results suggest that the final library should have two pixel plotting primitives: One for plotting single points (Version 1 of Table 1), and another for use by primitives that call upon the pixel writing routine a large number of times, e.g., line or arc drawing functions (Version 4 of Table 2).



The code for the five low-level pixel drawing routines used in our experiments is given here in an assembly language format that's compatible with the new Turbo C compiler. However, the routines also have been used successfully with other compilers after making the proper changes in segment names. Code is also given here for a Turbo C-compatible assembly language routine for drawing lines. This is excerpted from a much larger portable graphics library we have developed for use in our undergraduate graphics programming courses at the University of Pittsburgh. The article concludes with a brief discussion of the design of this library.

Overview of Results

Table 1 summarizes the timing results obtained for five different versions of an assembly language routine called *egadot()* when used in a test program developed with Turbo C.

The test program to evaluate the pixel drawing routines (Listing 1) draws a succession of 35 horizontal bars, each 10 pixels high, with the colors of the bars cycling through 16 colors. Since every pixel available on the screen is drawn by the function *egadot()*, a total of 640×350 or 224,000 pixels are plotted. After filling the screen with this pattern, the program then plots the sine function twice using the XOR operator. The second plot thus erases the first, restoring the original background.

The times in Table 1 were obtained with an Everex 8-MHz AT compatible, without a math coprocessor, using Turbo C. The slowest of the methods (Version 5, a call to the EGA BIOS *writedot* service) allows our benchmark program to write about 3,200 pixels per second, while the fastest (Version 4, which uses EGA write mode 2) permits

the benchmark test program to write about 22,000 pixels per second. These timings include the overhead associated with the C loops used to call *egadot()*. To find what this overhead is, the program starts out by using similar loops to call an empty function. On our machine, this took about 4.2 seconds. Subtracting the overhead gives the time for executing *egadot()* itself. The speed index column in Table 1 shows the reciprocal of these times divided by the time of the slowest version (the BIOS call). As you can see, Version 4 wins the prize in this category with a speed that's a little over 11 times that of the BIOS version. The writing rate for Version 4 without overhead is about 37,000 pixels per second. The dot routine with the Turbo C, Version 1.5, graphics library is 7100 pixels per second; and 5,600 pixels per second in the Microsoft Quick C library.

We'll discuss the difference in techniques used in the five versions of *egadot()* shortly. However, it will be useful first to discuss what *egadot()* does, and how this relates to the hardware associated with the EGA graphics system.

Design of the *egadot()* Procedures

The five versions of *egadot* discussed here have the ability to write "dots" (pixels) in any one of 16 colors, at any screen position (X,Y) where X lies between 0 and 639 and Y lies between 0 and 349. All of the procedures also give the programmer the option to write the pixels with either a SET or XOR operator. SET uses the information describing the new pixel's color to overwrite the color information for the previous pixel at the designated position. It can be considered the normal operator for writing new information to the graphics memory.

XOR combines the color information for the new pixel with that of the existing pixel. XOR has the valuable property that, if used twice at the same position with the same color, the new pixel is drawn and then erased, thus restoring the original background information. By repeating this double XOR operation at successive points, valuable animation effects can be obtained. For example, rubber-band lines, moving cursors, cross-hair pointers, and various "sprites" can be moved around the screen without disturbing the background.

Although there would be a slight speed advantage to writing separate functions for these two operators, the approach used here is to have only one set of primitives, with XOR being invoked by adding "256" to the color. Thus *egadot(x,y,3)* means "draw a pixel at position (x,y) with color 3 using SET to overwrite the old color," while *egadot(x,y,3+256)* means "draw a pixel at position (x,y) with a color that is the result of exclusive-oring color 3 with the existing color."

Writing Pixels to EGA Memory

The original EGA defined by IBM was based on the use of displays that allowed a pixel resolution of 640 horizontal by 350 vertical positions, with up to 16 colors available for each pixel. The EGA card contains enough RAM to store all of this information. This memory is called VRAM (Video RAM), and it is organized as four planes of approximately 32K each. The reason for the four planes is to provide 16 colors (which is the number of ways four bits can be written). The user can think of each colored pixel displayed on the screen as being backed up by four bits, with one bit in each color plane. (IBM calls these planes "maps".) These four bits are all in the same relative position within four different bytes of VRAM. However, some nice tricks in the hardware design allow the user to think of these four different bytes as all being at the same address. For the EGA, this address can be expressed as a 16-bit offset with respect to the video base segment address A0000 hex.

Programmers shouldn't have to think in terms of this VRAM organization, however. Rather, it's the screen that counts, and a graphics programmer can say "write a pixel of color *c* at position *x,y*," where *c* is one of 16 colors, *x* is the horizontal position of the pixel measured from the

left of the screen, and *y* is the vertical position of the pixel measured from the top.

The low-level functions take care of translating this data to actual VRAM addresses by using the following formulas:

Address offset of bytes corresponding to pixel at X, Y
 $= Y*80 + \text{int}(X/8)$

Bit position within these bytes corresponding to X =
 $X \bmod 8$

The macros in Listing 2 contain 8086 code for carrying out these calculations.

Use of the EGA Registers

Programmers familiar with the earlier CGA graphics system will know that the trick to making something appear on the graphics screen is to write the appropriate information to VRAM. That's still true for the EGA, but writing to the appropriate color planes is controlled by information you load into specific registers on the EGA board. There are actually three ways in which the EGA registers can be used to write a pixel, and these correspond to the way in which Versions 1, 2, and 3 of *egadot()* are coded. [Note that Version 4 uses the same registers as Version 3, but it removes the special initialization code needed for Version 3 from *egadot()*, assuming that this initialization will be taken care of in the calling program. In our test program, initialization is handled by uncommenting the lines using the *egareg()* function.] Version 5 calls the EGA BIOS *writedot* function, where the same registers as used in our Version 2 are employed behind the scenes.

The EGA registers are accessed through adjacent port addresses. The first of these pairs (called the address register) is loaded with a number called the *index*. The index determines which register of a group of registers sharing that address will be used. The second of the port pairs is then loaded with data appropriate to the function performed by the selected register. There are about 60 such functions available, so a complete description of usage of all the registers would take a small book (called the *EGA Technical Reference Manual*). However, the pixel-writing routines described in this article use only a handful of these functions, and discussion of two of them will shed light on their usage.

The principal port-pair addresses we'll use in our code are *03ce* paired with *03cf*, and *03c4* paired with *03c5* (all addresses are given in hex). The first of these pairs is used to access nine registers that are part of the EGA circuitry, called the "graphics-controller" by IBM. Which of these nine registers (and corresponding function) is selected is determined by the index value (0-8) loaded into port *03ce*. Thus, one can say that port address *03ce* "maps" into nine EGA registers. For example, sending the index value 3 to port *03ce* maps to the register associated with the function "data rotate and logical opera-

Table 1. Speed Benchmarks

Version	<i>egadot</i> total overhead time = time-time (sec)	Speed Index	Principal Technique Used
1	8.90 = 13.13 - 4.23	7.51	Use write mode 0 to send color via set/reset regs 3CF.00 & 3CF.01
2	8.95 = 13.18 - 4.23	7.47	Use write mode 0 to send color via color mask reg 3C5.00
3	8.24 = 12.47 - 4.23	8.11	Switch to write mode 2 with 3C5.05, write color to VRAM, switch back
4	5.99 = 10.22 - 4.23	11.16	Same as ver 3 but assumes write mode switching done by caller
5	66.84 = 71.07 - 4.23	1.00	Use BIOS INT 10H with service 0ch (<i>writedot</i>)

This table summarizes timings produced by the *test.c* program using five different versions of an EGA pixel drawing routine. Times in column 2 are given in seconds.

Listing 1. Test program used to obtain results of Table 1. Commented code using function `egareg()` is needed only when using version 4 of `egadot()`.

```

/*test.c-Test egadot routines dot1.asm, dot2.asm, etc.*/
#include "stdio.h"
#include "math.h"
#include "conio.h"
main()
{
    double showtime(), sin(), xscale, phi, s1, s2, emptime;
    int n, col, row, rowmax, colmax;
    int done=0;
    setmode(16); /* ega 640 x 350 graphics, 16 colors */
    rowmax = 350; colmax = 640;
    while(!done){
        setmode(16);
        printf(" *** Stand by for calibration of time for\n");
        empty loop ***\n");
        s1 = showtime();
        for (col=0; col<colmax; col++)
            for (row=0; row<rowmax; row++)
                empty(col,row,(row/10)%16); /* empty function to
                time loop overhead*/
        emptime=showtime()-s1;
        printf(" Time for C prog to make 224,000 empty calls\n");
        printf(" = %f sec\n",emptime);
        printf(" Press any key to time 224,000 executions of\n");
        egadot();\n");
        getch();
        printf(" >>>>> Here we go\n");
        s1=showtime();
        /* egareg(0x03ce,0x5,0x2); UNCOMMENT TO TEST VER. 4*/
        for (col=0; col<colmax; col++)
            for (row=0; row<rowmax; row++)
                egadot(col,row,(row/10)%16); /* fill screen 35 hor
                bars, 16 col */
        /* egareg(0x03ce,0x5,0x0); UNCOMMENT TO TEST VER.4 */
        /* egareg(0x03ce,0x8,0xff); UNCOMMENT TO TEST VER. 4*/
        s2=showtime();
        printf(" Time for 224,000 executions of egadot() = %f\n");
        printf(" sec\n",s2-s1);
        printf(" Time spent in egadot() = %f sec\n",
            s2-s1-emptime);
        printf(" XOR test now draws sin over bkgnd; 2nd XOR\n");
        printf(" restores bkgnd\n");
        printf(" Press any key when ready\n"); getch();
        xscale=12.5664/640;
        for (n=0;n<2;n++) /* use XOR twice */
        {
            /* egareg(0x03ce,0x5,0x2); UNCOMMENT TO TEST VER.4*/
            for(col=0;col<640;col++) /* draw sin function */
            {
                phi=col*xscale;
                row=(int)(174-174*sin(phi));
                egadot(col,row,col%16+256); /* cycle thru 16
                colors + 256 for XOR */
            }
            /*egareg(0x03ce,0x5,0x0); UNCOMMENT TO TEST VER.4*/
            /*egareg(0x03ce,0x8,0xff); UNCOMMENT TO TEST VER.4*/
        }
        printf(" One more time? (y/n)\n");
        if(tolower(getch()) != 'y') done=1;
    }
    setmode(3);
}

empty(a,b,c)
int a,b,c;
{ /*This produces Turbo C warning that can be ignored*/
/* Following functions shown for completeness. Some C
Compilers may contain similar functions in libraries */

double showtime()
{
    unsigned h,m,s,f;
    double hd,md,sd,fd;
    double totsec;
    struct r {unsigned ax,bx,cx,dx,si,di,ds,es;} rin,rout;

    rin.ax = 0x2c00; /* DOS function #44 (2c hex) */
    int86(0x21,&rin,&rout);
    h=rout.cx>>8;
    m=rout.cx & 0x00ff;
    s=rout.dx>>8;
    f=rout.dx & 0x00ff;
    hd=(double)h; md=(double)m;
    sd=(double)s; fd=(double)f; /* convert to double
    precision floats */
    totsec=3600*hd + 60*md + sd + fd/100;
    return(totsec);
}

```

... listing continues

tor select" (this is actually two functions for the price of one). The notation *03cf.03* is sometimes used to indicate this function. Once the index value 3 has been sent, the byte of data sent to the adjacent port at *03cf* determines exactly what the selected functions do. In this particular case, bits 0, 1, and 2 of the data byte specify the count for the rotate-left function, while bits 3 and 4 determine which logical operator will be used in certain write operations, by using the scheme 00=SET, 01=AND, 10=OR, 11=XOR. Thus, the following assembly code could be used to access this register, specifying "no rotate" and "write with operator XOR."

```

mov dx, 03ceh
mov al, 03h
out dx, al
mov dx, 03cfh
mov al, 018h ;data bits = 00011000
out dx, al

```

The other pair of ports (*03c4* and *03c5*) used to access EGA functions connected with pixel plotting map to five registers in the EGA circuitry that IBM calls the "sequencer." Sending index values 0, 1, . . . 4 to port *03c4* selects one of these sequencer registers. For example, sending the index value 2 to port *03c4* selects the "color enable function." Subsequent writing of color data to port *03c5* then determines which of the color planes will be used in the next write to VRAM.

To further clarify the use of these registers, let's examine the code for Version 2 of *egadot()* (Listing 3.2). The listing starts with definitions of the code and data segments Turbo C uses with the small memory model. These definitions will have to be changed for other compilers. Also note that Turbo C requires an underscore at the beginning of external procedure names.

At BLOCK 1, the macro *getaddr* of *egamac.h* (Listing 2) is used to place three pieces of data into CPU registers: (1) the address offset of the byte(s) in VRAM to which information will be written is placed in BX; (2) the bit mask specifying which bit of this byte is to be altered is loaded into CH, and (3) the color value specifying which planes are to be altered goes into DI. (Recall that we are also using the eighth bit of DI to specify whether the XOR function is to be used).

At BLOCK 2, an index value of 8 is placed in the address register at *03ceh* indicating "the register at *03cfh* will now be used to hold the pixel mask." The mask itself is then loaded into this register at BLOCK 3.

At BLOCK 4, the index 3 is sent to the register with port address *3ceh* to mean "use the register at *3cfh* to specify the bit operator to be used" (SET or XOR in our case). To specify XOR, the manuals tell us to send *018h* to port *03cfh* (BLOCK 5), while *00h* should be sent to specify SET (BLOCK 6). Then the index 2 is loaded in register *03c4h* at BLOCK 7 to mean "use *03c5* to hold the color map mask." The actual color mask is loaded at BLOCK 8.

At this point, the EGA knows which bit should be altered—and in which bytes of VRAM this change should be made, since it also knows which color planes should be modified. The instruction at BLOCK 9 then causes this information to be written into the bytes with address ES:BX] in the proper color planes.

Testing *egadot()*: Code for Other Versions of *egadot*

The simplest way to try the test program—say with Version 2 of *egadot()*—is the following (assuming that the assembly language file containing Version 2 is called *dot2.asm*, and that you have *masm.exe* available):

- Step 1. Assemble *dot2.asm* to produce *dot2.obj* using the case-sensitive switch */mx*.
- Step 2. Compile *test.c* to produce *test.obj*.
- Step 3. Link the object files from the first two steps with the C standard library for your compiler, calling the executable file *td2.exe*.

Turbo C provides a very neat way to accomplish these three steps with one command line:

```
A> tcc -ms -etd2 dot2.asm test.c
```

Here *tcc* is the name of the command line version of Turbo C (there is also a fancier, "programming environment" version); the switch *-ms* means use the small memory model; and *-etd2* means put the executable code in the file *td2.exe*. The extension *.asm* on the file *dot2* will cause automatic invocation of the masm assembler (provided it's on your disk), while the *.c* on test will cause invocation of the Turbo C compiler followed by invocation of the Turbo linker. The proper libraries will be used by the linker provided they are stored on your disk system as explained in the Turbo C manuals. It's all very painless, and very quick.

- Step 4. Run the test program by typing *td2* after the system prompt:

```
A>td2
```

If all goes well, you should see a display of horizontal bars in 16 glorious colors, followed by the plotting and "unplotting" of two cycles of the sine function.

Listing 3.2, which shows the code for Version 2 of *egadot()*, can serve as a model for the other versions. Only the central code that is marked as being specific to Version 2 needs to be replaced. Listings 3.1, 3.3, and 3.4 show this central code for Versions 1, 3, and 4, respectively. It is also necessary to have the function *egareg()* available when using Version 4. This is most conveniently added at the end of Version 4 of *egadot()*, as shown in Listing 3.4. Listing 3.5 shows the code for the BIOS version as a stand-alone procedure, since it does not have anything in common with the other versions.

Comparison with Microsoft QuickC

Microsoft has released QuickC, which retails for the same price as Turbo C and includes a graphics library. The pixel writing function in this library is called *_setpixel(x,y)*. Color must be specified with a separate function called *_setcolor(c)*, but there is no provision for specifying that the XOR operator be used in plotting the pixel. In order to compare the performance of QuickC to the approach described in this article, we recoded the first part of the program *test.c* to use the Microsoft Graphics functions, then compiled it with QuickC. (The second part of the program couldn't be recoded because of the lack of XOR.) The times for QuickC were:

- Plot 224,000 pixels = 49.30 seconds
- Execute loop calling *empty()* function = 8.90 seconds
- Actually plotting pixels = 40.43 seconds

The overall performance of the QuickC program is about seven times slower than that of the best Turbo C plus assembler program. The speed index for actually plotting colored pixels with QuickC is 1.65 relative to the data shown in Table 1, whereas the best Turbo C plus

```
setmode(mode)
int mode;
{
    struct r {unsigned ax,bx,cx,dx,si,di,ds,es;} rin,rout;
    rin.ax = mode;
    int86(0x10,&rin,&rout);
}
```

Listing 2. File *egamac.h* contains macros to calculate byte address offset and bit mask for graphics systems that use 80 consecutive bytes of VRAM for each each row of 640 consecutive pixel positions.

```
;File egamac.h Also works with Tandy 2000 code.
;Address offset -> bx, bit mask in ch, & color in di
;
getaddr macro
;
; multiply y-coordinate by bytes per row
    mov ax,8[bp] ; move y-coord into ax
    mov cl,4 ; shift count of 4
    sal ax,cl ; x 16
    mov bx,ax ; bx now has y-coord x 16
    sal ax,1 ; x 32
    sal ax,1 ; x 64
    add bx,ax ; bx now has y-coord x 80
; add x-coordinate/8 to address offset
    mov ax,6[bp] ; get x coord
    mov cx,7
    and cx,ax ; cx=x mod 8 (for use below as cl)
    sar ax,1
    sar ax,1
    sar ax,1 ; / 8
    add bx,ax ; bx now has y x 80 + x/8
; set up bit mask, store color in di
    xor cl,7 ; cl = 7 - (x mod 8)
    mov ch,01h
    rol ch,cl ; cl = shifting count, ch = mask
    mov di,10[bp] ; stash color in di for later use
    endm
;
pushreg macro
    push di ; Turbo C uses di & si for reg variables
    push bp ; could have avoided use of di by getting
    mov bp,sp ; ing color later. Getting color now
    endm ; neater but extra push/pop costs time.
;
popreg macro
    pop bp
    pop di
    endm
```

Listing 3.1 This code should be substituted for central part of Listing 3.2 to obtain version 1 of *egadot()*. Statement "and es:[bx],dl" is a bit tricky; causes CPU read of location es:[bx] which tells ega to latch ("grab") data at this location from all four color planes and place it into EGA data registers. "And" also forces a CPU write to tell EGA to load masked version of this data from its registers into planes determined by color data in set/reset register.

```
start:
    dec dx ; dx = 03ceh
    xor al,al ; index = 0, select set/reset reg
    out dx,al
    inc dx ; dx = 03cfh
    mov ax,di ; send color to reg
    out dx,al
    dec dx ; dx = 03ceh
    mov al,01h ; index = 1, enable set/reset reg
    out dx,al
    inc dx ; enable it, dx = 03cfh
    mov al,0fh
    out dx,al
    and es:[bx],dl ; does read to latch regs, &
    ; write to send masked latch content
    ; to color planes
    dec dx ; dx = 03ceh
    mov al,01h ; index 1 for set/reset register
    out dx,al
    inc dx ; dx = 03cfh
    xor al,al ; put 0 in al
    out dx,al ; set/reset back to normal
cleanup:
    dec dx ; dx = 03ceh
    mov al,08h ; index 8 = bit mask reg
```



```

out    dx,al
inc    dx                ; dx = 03cfh
mov    al,0ffh          ; all bits ON to restore
out    dx,al            ; bit mask to full enable

```

Listing 3.2. Assembly language code for version 2 of egadot(). Explanations keyed to the sections marked BLOCK 1, BLOCK 2, and so on are given in the text.

```

;File dot2.asm Version 2 of egadot()
;
;Following segment names used by Turbo C with small model
;
_text      segment      byte public 'code'
dgroup    group        _data,_bss
assume
cs:_text,ds:dgroup,ss:dgroup
_text      ends
_data     segment word public 'data'
_d0       label        byte
_data     ends
_bss      segment word public 'bss'
_b0       label        byte
_bss      ends
_data     segment word public 'data'
_s0       label        byte
_data     ends
_text     segment      byte public 'code'
_text     ends
;Above could be placed in header file for inclusion here

_text     segment byte public 'code' ;Begin dot2.asm code
include egamac.h                    ;Contains macros - See
                                     ; Listing 2
public _egadot;Leading underscore needed by Turbo
;
;This routine plots point on EGA color graphics system
_egadot proc near
; >>>> BLOCK 1
pushreg          ;macro
getaddr          ;macro -- see Listing 2
mov ax,0a000h    ;Starting address for EGA VRAM
mov es,ax
; >>>> BLOCK 2
mov dx,03ceh     ;dx = 03ceh
mov al,08h       ;index 8 to select bit mask reg
out dx,al
; >>>> BLOCK 3
inc dx           ;dx = 03cfh
mov al,ch        ;get the pixel mask
out dx,al        ;put it in mask reg
test di,256      ;is xor operation requested?
jz noxor         ;no -- go to noxor
; >>>> BLOCK 4
dec dx           ;dx = 03ceh
mov al,03h       ;index 3 for function select reg
out dx,al
; >>>> BLOCK 5
inc dx           ;dx = 03cfh
mov al,018h      ;make bits 3 & 4 ON to
out dx,al        ;get the XOR function
jmp start
noxor:
dec dx           ;dx = 03ceh
mov al,03h       ;index 3 for function select reg
out dx,al
; >>>> BLOCK 6
inc dx           ;dx = 03cfh
xor al,al        ;make al = 0 so that we have
out dx,al        ;bits 3 & 4 OFF for SET function

;START OF CODE UNIQUE TO VERSION 2 OF EGADOT
start:
; >>>> BLOCK 7
mov dx,03c4h     ;dx = 03c4h
mov al,02h       ;index 2 for color enable reg
out dx,al
inc dx           ;dx = 03c5h
mov ax,0ffh      ;enable all planes to
out dx,al        ;get set for clear with write of 0
and byte ptr es:[bx],0 ;read to latch old data
;thenclear VRAM with write of 0
dec dx           ;dx = 03c4h
mov al,02h       ;index 2 for color plane enable reg
out dx,al
; >>>> BLOCK 8
inc dx           ;dx = 03c5h
mov ax,di        ;load color data to
out dx,al        ;select planes to be used
; >>>> BLOCK 9
mov byte ptr es:[bx],0ffh ;write to VRAM (bit
                           ; mask & 0ffh)

```

... listing continues

assembler version has a speed index of 11.16. It seems clear that using Turbo C with a set of optimized assembly language primitives can provide a low-cost, high-performance development system for graphics applications.

Application to Building a Complete Graphics Library

The timings of Table 1 show that there is not much difference between Versions 1, 2, and 3 of *egadot()*. However, Version 3 can cause some glitches to appear on the screen because of the constant switching between write modes, so it is not recommended. (Incidentally, the Microsoft QuickC *setpixel* function also produces such glitches.) Version 4 is the clear winner in the timing department, and it has no such problems. However, Version 4 is cumbersome to use in an application because of the need to change to write mode 2 (accomplished with a call to *egareg()* as shown in Listing 3.4) before and after use of *egadot()*. Fortunately, one seldom plots points directly from an application program. Higher level objects, such as lines and arcs, are more likely to be used, and in this case the line or arc function can handle the changes in write mode once before and once after the loop that plots points using write mode 2. Listing 4 shows how to do this for a line plotting routine called *rline()*.

The routine shown in Listing 4 is called "simplified" because it does not check for illegal x or y values. It is assumed that it is called by a higher level line function that first applies a clipping algorithm to the line being drawn. Also, display adapters other than the EGA are not supported.

For our computer graphics teaching labs at the University of Pittsburgh, we have developed a library of graphics functions that work with a variety of display adapters. When the EGA is used, the assembly language routine based on Version 1 of *egadot()* is invoked for plotting single points, but Version 4 of *egadot()* is automatically selected for drawing lines, circles, and so on. The calling line function incorporates a clipping algorithm that keeps all plotted points on the screen or within a selected viewport, but with less overhead than the more obvious approach of checking each point against boundary values.

We have also developed a system that automatically switches to similar low-level code for other machine architectures, including the AT&T 6300, the Zenith Z100, CGA clones, and the Tandy 2000. (The ideas behind this portable system are the work of Don Redick.) The user selects the desired machine from a menu, and from then on all the low-level functions are for that machine. Since higher-level functions (e.g., polygon, box, fill) call upon these low-level functions, graphics programs linked with our portable graphics library can be developed and tested on any of these machines.

Thomas Dwyer is Professor of Computer Science at the University of Pittsburgh. He is currently working on the development of an advanced educational PC/Workstation called the MUSE (Micro University System Explorer). Arthur Huang and John King are graduate students in the same department.

All the source code for articles published in Micro/Systems is available on an MS-DOS disk. To order, send \$14.95 to Micro/Systems Journal, 501 Galveston Drive, Redwood City, CA 94063; or call Tim at (415) 366-3600. Please specify the issue number. Source code is also available on CompuServe; type GO DDJ and check the data library, line 14.

Did you find this article particularly useful?
Circle number 3 on the reader service card.


```

cleanup:
    mov     dx,03ceh    ;dx = 03ceh
    mov     al,08h      ;index 8 = bit mask reg
    out     dx,al
    inc     dx           ;dx = 03cfh
    mov     al,0ffh     ;all bits on to restore
    out     dx,al       ;bit mask to full enable
;END OF CODE UNIQUE TO VERSION 2 OF EGADOT
    popreg
    ret
_egadot endp
;
_text ends
;
END

```

Listing 3.3. Code used to change egadot() Version 2 to Version 3. Uses write mode 2 in which instruction "mov es:[bx], al" sends pixel data directly to planes determined by al.

```

start:
    dec     dx           ;dx = 03ceh
    mov     al,05h       ;index 5 for write mode reg
    out     dx,al
    inc     dx           ;dx = 03cfh
    mov     ax,02h       ;switch to write mode 2
    out     dx,al
    mov     ah,es:[bx]   ;do CPU read location to latch
                        ;current data
    mov     ax,di         ;put color value into al
    mov     es:[bx],al   ;write masked data to planes
                        ;determined by al
    dec     dx           ;dx = 03ceh
    mov     al,05h       ;index 5 for write mode reg
    out     dx,al
    inc     dx           ;dx = 03cfh
    xor     ax,ax        ;switch back to write mode 0
                        ;(normal default)
    out     dx,al
cleanup:
    dec     dx           ;dx = 03ceh
    mov     al,08h       ;index 8 = bit mask reg
    out     dx,al
    inc     dx           ;dx = 03cfh
    mov     al,0ffh     ;all bits ON to restore
    out     dx,al       ;bit mask to full enable

```

Listing 3.4. Used in converting egadot() Version 2 to Version 4. Method used is same as in Version 3, except that switching to write mode 2 (and back to write mode 0) is done only once by calling program.

```

start:
    mov     ah,es:[bx]   ;do a CPU read
    mov     ax,di        ;load the color value
    mov     es:[bx],al   ;write to planes set by al
;END OF CODE UNIQUE TO VERSION 4 OF EGADOT

;PROCEDURE EGAREG USED IN CALLING PROGRAM FOR VER.4
    public _egareg
_egareg proc near
    push    bp
    mov     bp,sp
    push    ax
    push    dx
    mov     dx,4[bp]
    mov     al,6[bp]
    out     dx,al
    inc     dx
    mov     al,8[bp]
    out     dx,al
    pop     dx
    pop     ax
    pop     bp
    ret
_egareg endp

```

Listing 3.5. Writes dots to EGA graphics screen using service 0ch of BIOS interrupt 10h. Since BIOS uses bit 7 to select XOR, program first converts to our bit 8 scheme.

```

;dot5.asm (Version 5 (BIOS call) of egadot)
; Standalone routine - do not use as part of Listing 3.2
_text segment byte public 'code'
    assume cs:_text

```

```

;
    public _egadot
_egadot proc near
    push    bp
    mov     bp,sp
    mov     cx,4[bp] ;x-coordinate
    mov     dx,6[bp] ;y-coordinate
    mov     bx,8[bp] ;color info, xor signal (bit 8)
    xor     al,al     ;al = 0
    test    bh,01h    ;is xor operation?
    jz      noxor      ;no; go to noxor
    mov     al,080h    ;yes; set BIOS xor signal (bit 7)
noxor:
    add     al,bl      ;get color
    xor     bh,bh      ;page 0
start:
    mov     ah,0ch     ;write dot service
    int     10h        ;BIOS video interrupt
    pop     bp
    ret
_egadot endp
;
_text ends
END

```

Listing 4. Simplified assembly language routine to draw line using egadot Version 4.

;_rline (Excerpt from Pitt Portable Graphics Library)
; In actual library, _rline is called from a function
; named pline() which first does Cohen-Sutherland clipping.
; Hence there are no out-of-bounds checks needed in _rline

```

_data segment word public 'data'
_s8 label byte
isxor dw 0
xcolor dw 256
y3 dw ?
y4 dw ?
y5 dw ?
y6 dw ?
aa dw ?
bb dw ?
dels dw ?
delp dw ?
dele dw ?
delsx dw ?
delsy dw ?
delse dw ?
deldx dw ?
deldy dw ?
delde dw ?
_data ends
_text segment byte public 'code'

```

```

;Video Ram Segment Address for EGA type display adapters
video2 segment at 0a000h
video2 ends
;

```

```

ARGS equ 4 ;for use with small model in finding
;args on stack

```

```

PUBLIC _rline
; This routine draws line from (x1,y1) to (x2,y2) in
; the specified color using Bresenham's algorithm.
;
_rline proc near
;
    push    bp           ; save registers
    mov     bp,sp
    push    si
    push    di
; set up x and y
    mov     si,1         ;start with positive 1 for x update
    mov     di,1         ;start with positive 1 for y update
; find |y2-y1|
    mov     dx,ARGS+6[bp] ;get y2
    sub     dx,ARGS+2[bp] ;subtract y1
    jge     storey       ;skip if y2-y1 is non-negative
    neg     di            ;move in negative y direction
    neg     dx            ;absolute value of y2-y1
storey:
    mov     deldy,di;store y update for diagonal moves
; find |x2-x1|
    mov     cx,ARGS+4[bp] ;get x2
    sub     cx,ARGS[bp]   ;subtract x1
    jge     storex       ;skip if x2-x1 is nonnegative
    neg     si            ;move in negative x direction
    neg     cx            ;absolute value of x2-x1
storex:
    mov     deldx,si;store x update for diagonal moves
; sort |y2-y1| and |x2-x1|

```



```

    cmp     cx,dx      ;compare dels with delp. Skip if
    jge     setdiag    ;straight moves in x direction
    mov     si,0       ;if straight-vertical: kill x update
    xchg    cx,dx      ;and exchange differences
    jmp     storedelsxy
setdiag:
    mov     di,0       ;if straight-horizontal kill y update
; store dels,delp,delsx, and dely
storedelsxy:
    mov     dels,cx    ;change in straight direction
    mov     delp,dx    ;change in perpendicular to straight
    mov     delsx,si   ;x update in straight direction
    mov     dely,di    ;y update in straight direction
; get initial values for x and y
    mov     si,ARG5[bp] ; x-coordinate
    mov     di,ARG5+2[bp] ; y-coordinate
; compute initial value & increments for error function
    mov     ax,delp
    sal     ax,1        ; 2*delp
    mov     delse,ax    ; change if straight move
    sub     ax,cx       ; 2*delp-dels
    mov     bx,ax       ; initial value
    sub     ax,cx       ; 2*delp-2*dels
    mov     delde,ax    ; change if diagonal move
; adjust count
    inc     cx
; set to write mode 2 so Version 4 of egadot can be used
    mov     dx,03ceh    ;select write mode register
    mov     al,05h      ;with index 5
    out     dx,al
    inc     dx          ;make dx = 03cfh
    mov     ax,02h      ;load bits to select write mode 2
    out     dx,al
    mov     dx,ARG5+8[bp];get color
    cmp     dx,xcolor   ;was 256 added to color?
    jle     lineolop    ;no, goto lineolop & use SET op
    sub     dx,xcolor   ;yes, get real xor color info
    mov     isxor,1     ; and set flag 'isxor' = 1
;
lineolop:
    call    egadot4     ;version 4 of egadot
    cmp     bx,0        ;determine straight or
    jge     diagonal    ;diagonal move
; case for vert or horiz straight line
straight:
    add     si,delsx     ; update x
    add     di,dely     ; update y
    add     bx,delse    ; update error term
    loop   lineolop     ; next point
    jmp     lineexit
; case for diagonal line
diagonal:
    add     si,deldx     ;update x
    add     di,deldy     ;update y
    add     bx,delde    ;update error term
    loop   lineolop     ;next point
lineexit:
    mov     isxor,0      ;reset flag 'isxor' = 0
    mov     dx,03ceh    ;select write mode register
    mov     al,05h
    out     dx,al
    inc     dx          ;dx=03cfh
    xor     ax,ax       ;ax=00h to put write mode back to 0
    out     dx,al
    dec     dx          ;dx = 03ceh
    mov     al,08       ;select bit mask register
    out     dx,al
    inc     dx          ;dx = 03cfh
    mov     al,0ffh     ;bit mask back to normal
    out     dx,al
rexit:
    pop     di          ;restore registers
    pop     si
    pop     bp
    ret
_rline endp
;
_text ends
END

```

End Listing 4

#1 C Interpreter

C-terp

Develop C programs with the safety, and convenience that only our highly acclaimed C interpreter can provide.

"C-terp is an excellent way to develop C programs. It has a good user interface and never went astray.

Ken Pugh, The C Users' Group

"C-terp comes closest to this ideal [C development environment]"

John Unger, BYTE

"... easy to use, powerful, and a time saver ... one of my favorite production tools"

Larry Jordan, The C Journal

"C-terp is a high-power interpretive environment ... an excellent product."

Computer Language

Not just a learning tool: large memory model, multiple-modules, a multi-file, reconfigurable editor, automatic make, virtual memory option, shared symbols option, complete object code compatibility, links in your compiler's entire library.

Full screen debugging: single-step; side-step; display variables, arrays, structures; watch expressions; watch conditions; execute any C expression (even with macros); traceback; leave-function; sticky, temporary and cursor- directed breakpoints. Out-of-bounds pointers are trapped.

Even Bill Gates admits that there's nothing like an interpreter to enhance your productivity and this is particularly true for C.

Gimpel Software

3207 Hogarth Lane
Collegeville PA 19426
(215) 584-4261

For MS-DOS, specify compiler

\$298.00 Microsoft 4.0 / 5.0, Lattice, Aztec,
C86, C86 Plus, Mark Williams
\$139.00 Turbo C 1.0 / 1.5
\$498.00 Xenix 286 / 386

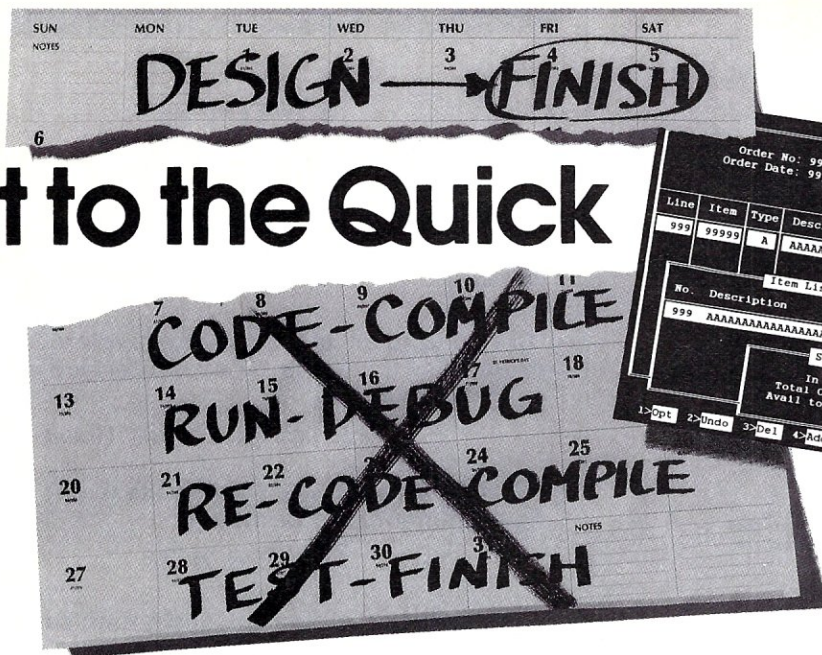
Order Today - MC, VISA, COD

PA residents add 6% sales tax. Outside USA add \$20.
C-terp is a trademark of Gimpel Software.

CIRCLE 75 ON READER SERVICE CARD

Attn
Btrieve®
programmers:

Cut to the Quick



Order No: 999
Order Date: 99/99/99

Customer No: 99999
Address: AAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAA

Line	Item	Type	Description	Quantity	Unit Price	Total Price
999	99999	A	AAAAAAAAAAAAAAAAAAAAAAAAAAAA	-9.999		

Item List

No.	Description	Type	Price
999	AAAAAAAAAAAAAAAAAAAAAAAAAAAA	A	-999.999

Stock Status

In Stock: -999.999
Total Orders: -999.999
Avail to Sell: -999.999

Order Sum
Discount: -999.999.99
Sub-Total: -999.999.99
Sales Tax: -999.999.99
Order Total: -999.999.99

1>Opt 2>Undo 3>Del 4>Add 5>Zoom 6>Expr 7>Draw 8>Task 9>End 10>Help

MAGIC PC ELIMINATES CODING ... CUTS MONTHS OF DATABASE DEVELOPMENT!

Time is money. And coding a DBMS application like Accounting or Order Entry takes a lot of both. Simply because hacking out mountains of code with your RDBMS or 4GL is too slow. Not to mention the time to re-write if you make a mistake or change the design.

EXECUTION TABLES ELIMINATE CODE!

Magic PC cuts months of your application development time because it eliminates coding. You program with the state-of-the-art Execution Tables in place of conventional programming.

HOW DOES IT WORK?

Magic PC turns your database design scheme directly into executable applications without any coding. Use Execution Tables to describe only what your programs do with compact design spec's, free from lengthy how to programming details. Each table entry is a powerful non-procedural design instruction which is executed at compiled-like speed by a runtime engine. Yet the tables can be modified "on the fly" without any maintenance. Develop full-featured multi-user turn-key systems with custom screens, windows, menus, reports and much more in days — not months! No more low-level programming, no time wasted...

MAGIC PC™

The *Visual* Database Language



"Magic PC's database engine delivers powerful applications in a fraction of the time... there is no competitive product."



"Overall, Magic PC is one of the most powerful DBMS packages available."

- Quick Application Generator
- BTRIEVE® — based multi-user RDBMS
- Visual design language eliminates coding
- Maintenance-free program modifications
- Easy-to-use Visual Query-By-Example
- Multi-file Zoom window look-ups
- Low-cost distribution Runtimes
- OEM versions available

ATTENTION BTRIEVE® USERS

Now you can quickly enhance your BTRIEVE®-based applications beyond the capabilities of XTRIEVE® and RTRIEVE®. Use Magic PC as a turn-key BTRIEVE® Application Generator to customize your applications without even changing your existing code.

AKER

19782 MacArthur Boulevard, Suite 305
Irvine, California 92715

TLX: 493-1184

FAX: 714-955-0199

DATABASE PROGRAMMERS

Join the thousands of professional database programmers and vertical market developers who switched to Magic PC from dBase®, R:BASE®, Paradox®, Clipper®, Dataflex®, Revelation®, Basic, C, Pascal, etc.

TRY BEFORE YOU PAY

We're so sure you'll love Magic PC — we'll let you try the complete package first. Only a limited quantity is available, so call us today to reserve your copy. Pay for Magic PC only after 30 days of working with it. * To cancel... don't call... simply return in 30 days for a \$19.95 restocking fee.

OR PAY NOW AT NO RISK

Pay when you order and we'll wave the \$19.95 restocking fee so you have absolutely no risk.

SPECIAL OFFER \$695

\$199



Magic LAN multi-user — \$399

Magic RUN — call for price

**Order Now Call:
800-345-MAGIC**

In CA 714-250-1718

MS

Add \$10 P&H, tax in CA. International orders add \$30.

*Secured with credit card or open P.O. Valid in US.

Dealers welcomed

CALL ADVERTISER DIRECTLY

Microsoft Windows/386

*A robust environment for running multiple
DOS and Windows applications.*

by William Wong

Windows/386 is a special version of Microsoft Windows for 80386-based DOS systems containing at least two megabytes of memory. It runs DOS, Windows 1.0x, and Windows 2.0x applications. Windows/386 takes advantage of some of the power of the 80386 by using the Virtual 86 mode and thus allowing the PC to use more memory than DOS by running multiple applications. Each application is limited by the DOS limit of 640 kilobytes. Windows/386 does not enable applications to access the 80386's protected mode, although Windows/386 itself utilizes protected mode. Windows/386 does effectively utilize the 80386 by running multiple existing DOS and Windows applications concurrently, something OS/2 even with the Presentation Manager will not do.

Multitasker alternatives to Windows/386 include DESQview (reviewed in *Micro/Systems*, February 1988), PC-MOS/386 and Concurrent DOS 386 (both reviewed in *Micro/Systems Journal*, November/December 1987), VM/386, Windows 2.0, and the OS/2 Presentation Manager. DESQview is a multiwindow multitasker with 80386 Virtual 86 mode support. DESQview can even run Microsoft Windows 1.04 in one of its windows and hence can support a Windows application. It is the closest competitor to Windows/386. VM/386 (see page 60) is similar to DESQview but does not support Windows as an application. PC-MOS/386 and Concurrent DOS are complete DOS-compatible multiuser/multitasking operating systems that can also be used as single-user systems. They provide multiple screens, but only one is visible at a time. Neither supports Microsoft Windows.

Windows 2.0 does not take advantage

of the 80386, but it can run multiple DOS and Windows applications using LIM 4.0 EMS RAM. It can be run on an 80386 with extended memory by employing a third-party virtual-control program that uses the Virtual 86 mode to simulate EMS RAM. This approach is similar to Windows/386, but it does not allow Windows 2.0 to run a DOS application within a Virtual 86-mode partition (which provides significant protection mechanisms to Windows/386). The OS/2 Presentation Manager (PM) will look like Windows/386, although the front-end program for PM should be better. However, PM will not run any Windows applications, and only one DOS application can be run in its "compatibility box." OS/2 uses only the 80286 protected mode, even when running on an 80386 system.

So why all the interest in the variety of alternatives? One element of the puzzle is the increasing number and complexity of Windows applications, which is making Windows into a viable alternative work environment. Another element is the influence of OS/2 PM, which will look a lot like Windows/386. Finally, there is the matter of timing. OS/2 is out, but PM will not be out until the fourth quarter of 1988, and it will not take advantage of the 80386 to run Windows applications or multiple DOS applications. It will be years before we see a 386 version of OS/2. Thus, Microsoft needed a product that would provide DOS and Windows compatibility while we await (for lack of a better name) OS/3.

Windows/386 does have some problems, but nothing major. It is possible to crash it using older Windows 1.0x applications and DOS applications. Programs that play with DOS or with disks, such as disk

reorganizer programs, also are likely to cause problems. However, Windows/386 can usually recover from most DOS application failures, and most older Windows applications are being recompiled for Windows 2.0 and Windows/386.

How It Actually Works

Windows/386 is actually a virtual-monitor program, running a special version of Windows 2.0 as its user interface. The virtual monitor runs in protected mode. It provides support for multiple Virtual 86-mode partitions, as well as LIM 4.0 EMS support that can be used by DOS applications such as Lotus 1-2-3 Version 2.0 or Windows.

The Virtual 86 mode of the 80386 allows a program to run in a logical, one-megabyte address space. The 80386 supports a paged-memory mapping system from protected mode, allowing the logical address space to be mapped to the physical address space accessible by the 80386. This allows a program running in the Virtual 86 mode to use extended memory that is above the 1 MB of conventional memory normally used by DOS applications. The logical address space is divided into a fixed number of pages hidden from the application. The 80386 supports page sizes that are a power of two (e.g., 1 or 4 kilobytes). A page can be mapped to any physical page of memory, and multiple logical pages can be mapped to the same physical page. Logical pages can also be marked as inaccessible, causing a system trap when the page is accessed. It is possible to implement a virtual memory system using this support, but Windows/386 does not take advantage of this feature at present.

The mapping process allows three things to happen. First, each applica-

tion running in a Virtual 86-mode partition gets its own memory for the application program and data. Second, each application partition has access to a mapped copy of DOS. Third, each application partition gets a copy of all the hardware buffers it will use, either directly or indirectly. The latter allows applications to share display adapters. The application will have its display adapter buffer pages mapped to the appropriate spot when using the full screen, but an application will have these same pages mapped to an area of memory when another application is using the full screen. In either case, an application will access the memory buffer in the way it expects to, and the hardware transparently maps buffer pages to the appropriate spot. The operating system will swap between areas of memory when the application associated with the screen changes. In a windowed environment, each application has its video display buffer area mapped to a memory area, and the virtual monitor copies the visible areas from memory to the display adapter's buffer.

The Virtual 86 mode also controls access to I/O locations, allowing full access, no access, or limited access

to I/O. It does not remap access as it does with memory pages, but the 80386 can keep track of what I/O operations are performed so that the virtual monitor can handle and simulate the operations. For example, the virtual monitor may access the serial port for an application. This approach is slower than allowing the application to do the job directly, but it does prevent the application from causing problems by controlling the hardware. The mapping support also simulates EMS RAM by using extended memory mapped to the address space of a program running in Virtual 86-mode.

The 80286 does not have a Virtual 86 mode which is one reason why it is often said to be "brain-damaged." The lack of a Virtual 86 mode prevents OS/2 from running more than one DOS application at a time. OS/2 runs a DOS application in normal mode, using the first 640K of RAM (called the "compatibility box") instead of using the 80286 protected mode. Normal mode does not have the protection provided by the 80286/80386 protected mode or the 80386 Virtual 86 mode.

WIN386.EXE is the Windows/386 virtual monitor program and is loaded first. The virtual monitor sets up a

Virtual 86 session and loads the WIN386.EXE program, which is a special version of Windows 2.0, and runs just like the Windows 2.0 version, with two important differences. The first is the simulated EMS support, and the second is the ability to start off DOS applications with their own memory within Virtual 86-mode sessions. These are services provided by the virtual monitor. The virtual monitor also reports the status of DOS applications and provides access to the screens used by DOS applications, thus allowing character-based cut-and-paste operations between Windows and DOS applications. The link between the virtual monitor and Windows 2.0 is transparent to both Windows and DOS applications.

The amount of space reserved for DOS applications is specified within the Windows initialization text file, WIN.INI, or an application can have its requirements specified through a .PIF file. Both mechanisms are used by Windows 1.0x and Windows 2.0 as well as Windows/386. Each application runs within its own memory space. If the application requires EMS support, it is mapped to its memory space by the virtual monitor.

Unlike DOS applications, Windows

Attention Graphics Programmers and System Developers

The TI 34010 is the best news to come along in the graphics processor arena in a number of years. PC Tech, Inc. has developed two TMS 34010 based PC compatible display adapters that allow the full power of the TMS 34010 to be used on PC/XT/AT compatible machines. DGIS is available for either board.

MONOCHROME

- 256k display memory
- 256k program memory
- Up to 48 MHz 6 MIPS operation
- Both ECL and TTL monitor drivers
- Multiple programmable dot clock frequencies
- Typical resolution 736 x 1024
- Capable of drawing 125 million pixels per second
- CGA / MDA / Hercules Emulation
- Suggested list price \$595 (\$1295 with full page black on white monitor.)

COLOR

- Up to 1 Mbyte display memory (8 bit / pixel)
- 1 - 4 Mbyte program memory
- 256 simultaneous colors from palette of 256,000
- Up to 48 MHz 6 MIPS operation
- Dual (piggyback) single slot board
- Multiple programmable dot clock frequencies
- Typical resolution 1024 x 768
- Custom dot clocks available
- Suggested list price \$1295 (\$1295 video memory, 1 Mbyte program memory)

TI 34010 Assembler, Debugger and "C" Compiler available!

Both boards support DGIS from Graphic Software Systems, Inc. Most major text and graphics applications are supported on both boards via DGIS and low level graphics interfaces. Custom implementations of either design and private labeling are also available.

Contact: PC Tech, Inc.
904 N. 6th Street
Lake City, MN 55041
612-345-4555
FAX: 612-345-5514

PC, XT and AT are trademarks of International Business Machines Corp.
DGIS is a trademark of Graphic Software Systems, Inc.
Prices subject to change.

CIRCLE 46 ON READER SERVICE CARD

A Reliable PC/XT Compatible for the Cornerstone of Your Products

SLICER Announces The SLY40-XT.

The SLY40-XT is a small (4-1/4" by 9-1/4") four layer card featuring all of a PC/XT mother board functions. This board is composed of just 17 low power CMOS ICs, and it simply plugs into a passive back plane or SLICER's TEN SLOT BUS BOARD.

- NEC's 8 MHZ V40
- One Megabyte of Zero Wait State RAM
- Ideal For Tough Industrial, OEM and Portable Applications
- American Made and Fully Supported by Slicer

PC and XT Are Trademarks of International Business Machines

MasterCard
Visa
Check
Money Order
C.O.D.



SLICER Computers Inc.
3450 Snelling Ave. So.
Minneapolis, MN 55406
612/724-2710
Telex 501357
SLICER UD

CALL ADVERTISER DIRECTLY

applications do not run in their own memory space. Instead, all Windows applications run in the memory space of the single Windows 2.0 program that is run under the virtual monitor. The big advantage is in the EMS support provided by the virtual monitor. Each Windows application is essentially mapped to EMS RAM and swapped out as necessary. This approach is necessary because the Windows applications can share memory and share the Windows services. DOS applications are isolated and do not share memory among themselves or with Windows. So, for Windows applications, Windows 2.0 with full LIM 4.0 EMS support is no different from Windows/386, except that Windows/386 must run on an 80386 system.

This approach to the partitioning of program memory shows why Windows/386 is more robust than Windows 2.0 when running DOS applications, but it is just as susceptible if a Windows application fails.

It is possible to crash Windows/386 from a DOS application or a Windows application. However, the Virtual 86 mode provides significantly more protection for failing DOS applications—as well as for DOS applications that try to use memory

or peripherals that might cause Windows/386 to crash.

The virtual monitor provides preemptive multitasking support for Windows and for any DOS applications that are running. DOS applications

Each Windows application is mapped to EMS RAM and swapped out as necessary.

can be set to run in exclusive mode, in which case the DOS application runs until it terminates or until the user suspends the application to return to Windows. Windows/386 provides standard Windows non-preemptive multitasking.

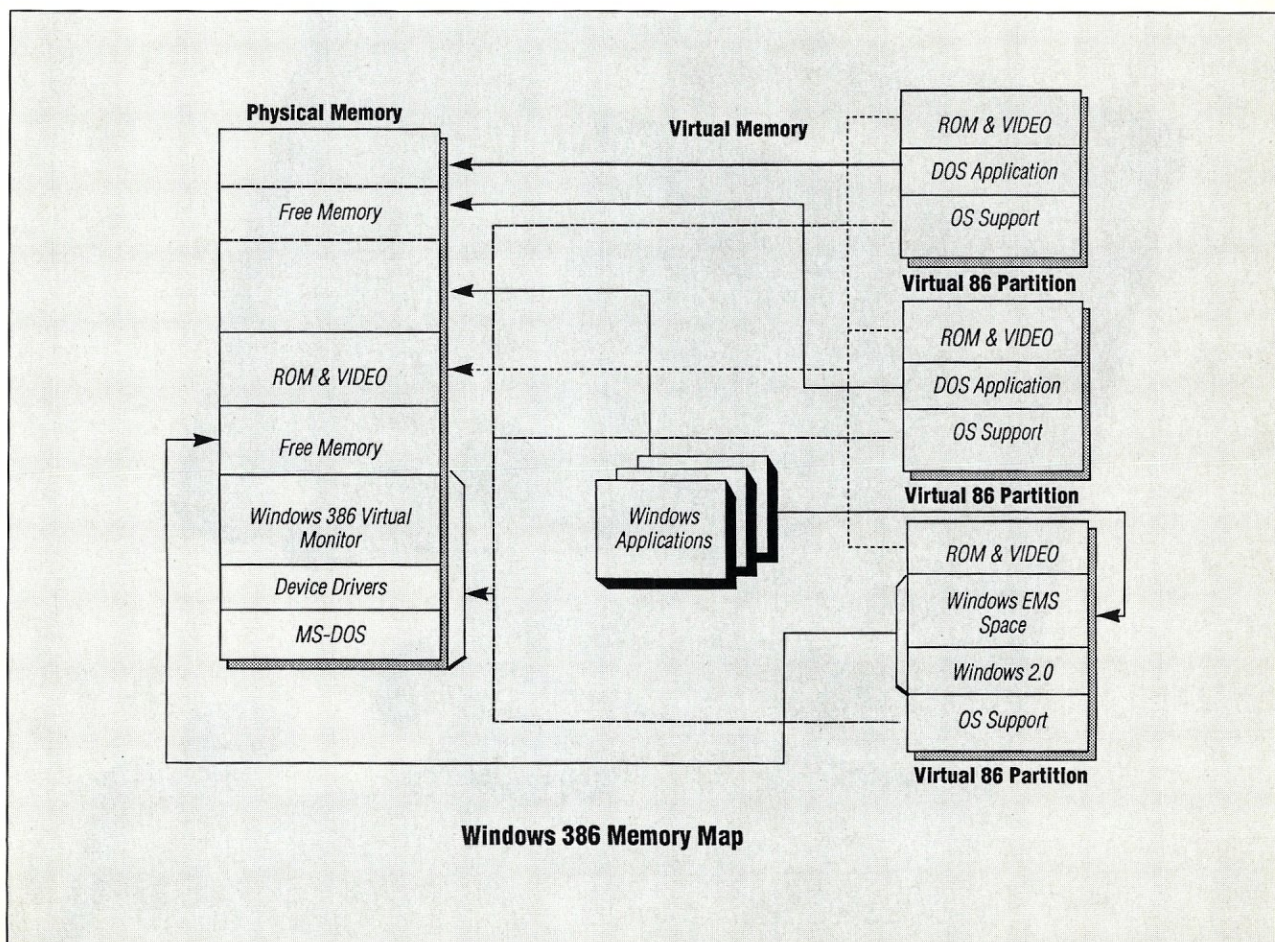
Severe performance degradation

will occur if several DOS applications are working at the same time, such as compiling or linking. However, even 10 copies of a spreadsheet will cause little performance degradation if they are all waiting for keyboard input.

Compute-processing performance for a single application does not really change much because the multitasking overhead is relatively low. Disk-access performance is essentially the same, but screen-update performance is somewhat slower, depending upon the application. Character-mode applications suffer the most because Windows/386 runs in graphics mode. Windows-application screen update is similar to running Windows 2.0 on an 80386, but it can be faster, depending upon the implementation of the screen device drivers.

A Look at the Documentation

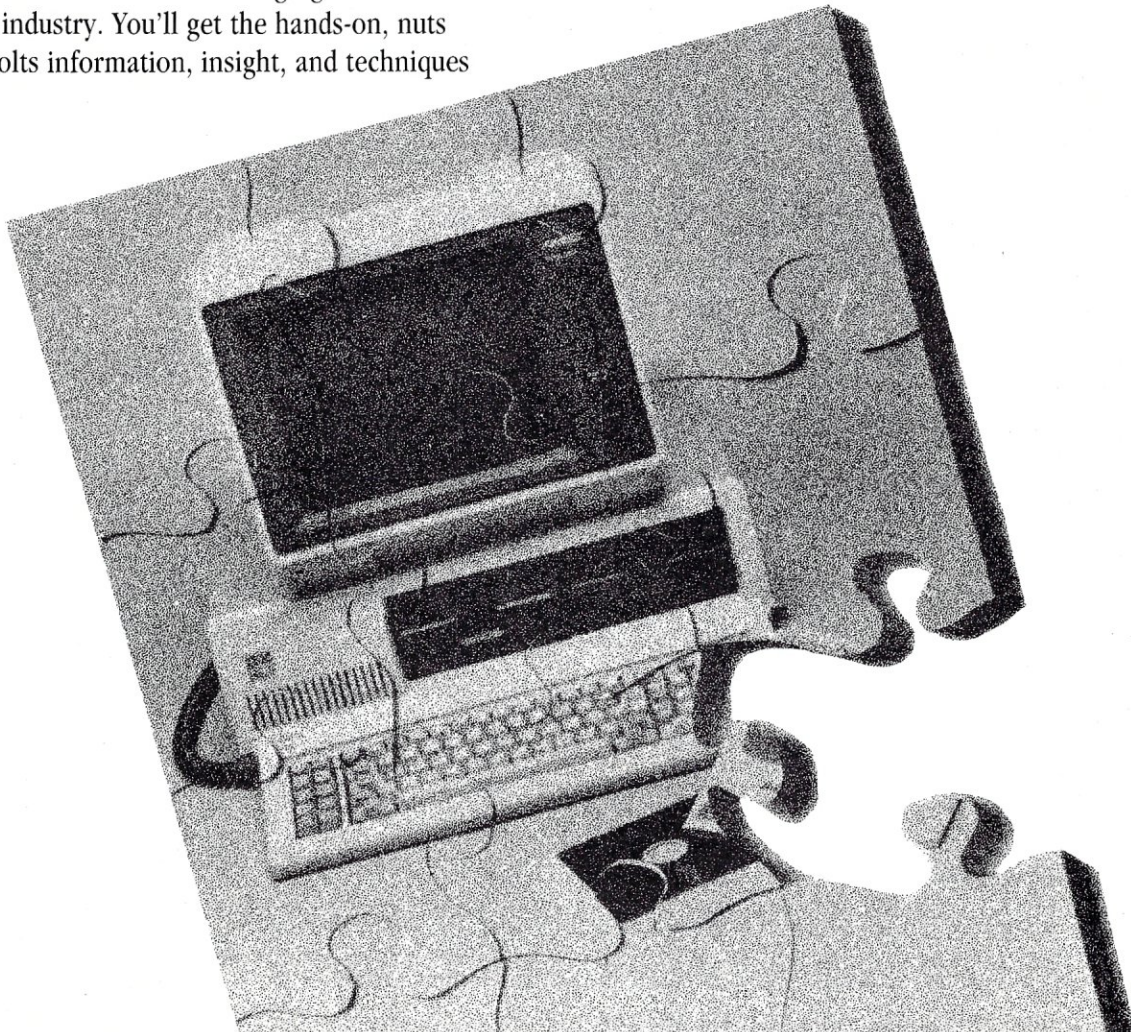
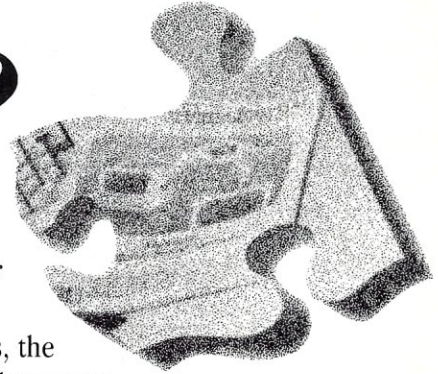
Windows/386 comes with the same documentation as Windows 2.0, plus a small booklet entitled *Using Windows/386*. It covers installation, getting started, and running DOS applications, including a discussion of .PIF files. Considerations such as display device drivers for Windows/386 are not covered. Also lacking is a good



Why puzzle when you don't have to?

Micro/Systems Journal has the answers. Whether it's networking, systems integration, programming, or scientific computing questions, *M/SJ* will lead you out of the maze of microcomputer mayhem. With each issue you'll find comprehensive coverage of all the technical information that will keep you up-to-date with the ever-changing microcomputer industry. You'll get the hands-on, nuts and bolts information, insight, and techniques

that *M/SJ* is famous for providing . . . in-depth tutorials, reviews, hints, the latest on multitasking, languages and operating systems. So stop your puzzling . . . subscribe right now and the answers will be yours. Simply drop the attached card in the mail—that's all there is to it.



technical overview that could clear up some confusion about why some programs do not run. Additional error messages are listed, but the explanations and solutions are too brief.

The Windows 2.0 documentation covers the operation of Windows 2.0 and the standard set of Windows applications, including Windows Write, Windows Paint, a terminal program, and utility programs. The Windows 2.0 documentation does not mention Windows/386. The Windows 2.0 documentation is well written and examples are included.

What Runs Under Windows

Most DOS applications will run under Windows/386. However, in order to run properly, certain applications may require a special Program Integration File (PIF). For example, some applications will only run in full-screen mode, where Windows hides in the background. Windows is selected using the Alt+Tab or Alt+Esc keys. DOS applications can also be run in exclusive mode, and can be forced to suspend when not active.

Terminate-and-Stay-Resident (TSR) applications should not be loaded before Windows/386 is loaded. In-

stead, they should be loaded from a DOS partition within Windows/386. This must be done by first running COMMAND.COM and then loading the TSR. The TSR is available by selecting the window running COMMAND.COM and using the appropriate hot key sequence to access the TSR. Other applications can be loaded from the COMMAND.COM partition, and the TSR will be accessible while these applications are running. Multiple COMMAND.COM windows can be run, but TSRs must be loaded into each window if they are to be accessible.

Windows 2.0 applications should run on Windows/386 without any problems since both are essentially the same operating environment. Windows 1.0x applications will also run under Windows/386 if they run under Windows 2.0. Fonts used by Windows 1.0x must be converted for use under both Windows 2.0 and Windows/386.

Windows/386 will not support any protected-mode DOS extenders, such as those provided by Phar Lap or A.I. Architects. It will not run applications that use this type of DOS extender—including Golden Common Lisp, although the latter can run if it does not use extended memory.

Windows/386 comes with an extended-memory RAM disk driver and a disk-caching program, SMARTDRV, which are the only ones that work with Windows/386.

Device drivers are also a general problem with Windows/386, because they must be specially written for Windows/386 according to the type of driver. In general, with the exception of extended-memory RAM disks, disk device drivers work well. Most printer and network device drivers work, but there are some exceptions. Windows/386 does not work with Novell's Advanced NetWare 2.0a, but will work with an updated version of NetWare. Windows/386 works with Banyan VINES for disk and printer services, but problems can occur if multiple VINES applications are run as DOS applications. Network device drivers sometimes have problems because of the multitasking nature of Windows/386.

Windows device drivers are a problem because Windows/386 requires special device drivers for the pointing and display devices. Most major pointing and display devices are supported, including, of course, the Microsoft mouse and a number of other popular mice. Display devices include the EGA and the VGA,

Interested in Becoming a PC Seminar Leader?

Are you:

- ✓ an expert in your field
- ✓ an independent consultant
- ✓ experienced in teaching today's professionals
- ✓ knowledgeable in real-world applications
- ✓ able to travel

Data-Tech Institute, a leader in continuing education is currently looking for PC experts to develop and teach seminars on the following topics:

- | | |
|--------------------|---------------------|
| — XT, AT, and PS/2 | — PC Integration |
| — OS/2 and DOS | — Networking |
| — LANs | — PC Systems Design |

Qualified individuals please send resume and credentials to:

**DATA-TECH
INSTITUTE**

DATA-TECH INSTITUTE
Lakeview Plaza
P.O. Box 2429
Clifton, NJ 07015
Attn: Sandy Marschall

CIRCLE 91 ON READER SERVICE CARD

• 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 • 386 •

NOW!

C → 386

Pascal

Paradox 386
Foxbase+ 386
386-MATLAB, Weitek

... and others ...

These and other protected-mode 32-bit 80386 programs are among the first to take advantage of the full power of the 386. They and practically every 386 protected-mode MS-DOS program that's shipping were done with MetaWare's compilers.

It's no surprise. The recognized leader, MetaWare introduced the first C and Pascal compilers that generate protected-mode 386 code for running on any 386 MS-DOS machine (e.g., the Compaq 386 or the IBM PS/2-80) over a year ago. High C™ and Professional Pascal™ are well-established and proven.

Smart software developers aren't waiting! Industry leaders such as Borland (ANSA) and Fox use MetaWare's compilers to get dramatic increases in speed and functionality. Don't wait years for Microsoft's 386DOS—your competition will have a big jump on you!

Expand your application to the large 32-bit address space and the full 32-bit registers of the 80386. Go with the long-standing leader. Contact MetaWare for your 80386 software solution today!

(408) 429-6382 telex 493-0879

MetaWare™

INCORPORATED

903 Pacific Avenue, Suite 201 • Santa Cruz, CA 95060

The Clear Choice for Large Programming Projects —PC Tech Jnl

© 1987 MetaWare. Trademarks: Paradox 386, Ansa Foxbase+ 386, Fox Software 386-MATLAB, Mathworks, High C Professional Pascal, MetaWare, MetaWare.

CIRCLE 56 ON READER SERVICE CARD

but Hercules support is missing. However, this should not cause you too much worry. If you can run Windows 2.0 now, there will probably be a new device driver available from Microsoft or from the vendor of the pointing device or the display adapter. For example, the 800-by-600 pixel Super EGA adapter from Genoa will work as a normal EGA, and Genoa has promised a new driver for the 800-by-600 mode. Windows 2.0 device drivers do not work with Windows/386. The lack of some device drivers may force you to stick with Windows 2.0 until the appropri-

ate Windows/386 device driver becomes available.

Summary

Windows/386 provides a useful, robust environment for multiple DOS and Windows applications running on an 80386. It lets you utilize the power and memory capabilities of the 80386 today. It does not provide access to OS/2 applications or to the 80386's 32-bit protected mode.

DESQview is one alternative to Windows/386 that may be a better choice if you do not need to run Windows applications. DESQview is

faster than Windows/386 for character-mode applications such as most spreadsheets. (Microsoft's Excel is an exception in that it is a graphic Windows application.)

The OS/2 Presentation Manager is really not an alternative to Windows/386. OS/2 can only run a single DOS application, and multitasking applications must be written specifically for OS/2. And Windows/386 cannot run OS/2 Presentation Manager applications, since Windows/386 only runs DOS or Windows applications.

OS/2 with the Presentation Manager and Windows/386 will probably merge in a few years, as OS/2 is extended to run on the 80386 using

SERIOUS DEBUGGING AT A REASONABLE PRICE



All the speed and power of a hardware-assisted debugger at a software price

Hardware-level break points

REAL-TIME break points on memory locations, memory ranges, execution, I/O ports, hardware and software interrupts. More powerful break points than ANY software-only debugger on the market.

Break out of hung programs

With a keystroke - no external switch necessary. Even with interrupts disabled.

Breaks the 640K barrier

Soft-ICE uses ZERO bytes of memory in the first 1MB of address space.

Works with your favorite debugger

Soft-ICE can be used as a stand-alone debugger or it can add its powerful break points to the software debugger you already use.

Solve tough systems problems too

Soft-ICE is ideal for debugging TSRs, interrupt handlers, self booting programs, DOS loadable device drivers, non-DOS operating systems and debugging within DOS & BIOS. Soft-ICE is also great for firmware development because Soft-ICE's break points work in ROM.

How Soft-ICE Works

Soft-ICE uses the power of the 80386 to surround your program in a virtual machine. This gives you complete control of the DOS environment, while Soft-ICE runs safely in protected mode. Soft-ICE uses 80386 protected mode features, such as paging, I/O privilege level, and break point registers, to provide real-time hardware-level break points.

"It is a unique, effective solution to a wide range of critical debugging problems"
COMPUTER LANGUAGE -- April 1988

RUN CODEVIEW IN ONLY 8K

MagicCV



CodeView is a great integrated debugger, but it uses over 200K of conventional memory. MagicCV uses advanced features of the 80386 microprocessor to run CodeView in a separate virtual machine in extended memory. This allows MagicCV to run CodeView using less than 8K of conventional memory on your 80386 PC.

Don't let 640K be your limit!

If you are closing in on the 640K limit and would like the power of CodeView, MagicCV is for you.

Find those hidden bugs!

Even if you're not closing in on the 640K limit, running CodeView with MagicCV makes your debugging environment much closer to the end user's program environment. You can use CodeView to locate subtle bugs that only occur when there is plenty of free memory.

MagicCV is easy to use

If you are a CodeView user, you already know how to use MagicCV too. Just type MCV instead of CV, everything else is automatic.

MagicCV with Soft-ICE

Using Soft-ICE with CodeView gives you the features necessary for professional level systems debugging. MagicCV and Soft-ICE can work in concert with CodeView to provide the most powerful debugging platform you will find anywhere.

Buy Both and SAVE \$86

MagicCV	\$199
Soft-ICE	\$386

(603) 888 - 2386

Nu-Mega Technologies

PO Box 7607
Nashua, NH 03060-7607

30 day money-back guarantee
Visa and Master Card accepted

Both require 80386 PS/2 or AT compatible

As an extra bonus, by ordering both MagicCV and Soft-ICE together you SAVE \$86.

Codeview is a trademark of Microsoft Corp.

*OS/2 with the
Presentation
Manager and
Windows/386
will probably
merge in a few
years.*

the 32-bit protected and Virtual 86 modes. It is conceivable that this version of OS/2 will be able to run DOS applications, Windows applications, OS/2 applications, Presentation Manager applications, and 32-bit OS/2 Presentation Manager applications. However, it could be a long wait for the 80386 version of OS/2, and Windows/386 is here now. If you need to run 32-bit 80386 applications, I recommend that you also look into an 80386 UNIX from companies such as SCO, Microport, or Interactive Systems.

William Wong is president of Logic Fusion, Inc., a systems software development firm located in Morrisville, Pennsylvania.

Did you find this article particularly useful?
Circle number 4 on the reader service card.

Product Information

Windows/386 \$195

Microsoft Corporation

16011 NE 36th Way
Redmond, WA 98052
(800) 426-9400
(206) 882-8080

CIRCLE 103 ON READER SERVICE CARD

A Routine to Dump Graphics Screens to Laser Printers

by Jeffrey Ross

This memory-resident program allows you to manipulate an HP LaserJet to accept any kind of Hercules-driven graphics dump.

This article describes the development of a memory-resident screen print utility to print text or graphic images from a Hercules monochrome graphics board to a Hewlett-Packard LaserJet or compatible printer. The Hercules board has become a *de facto* standard for high-resolution monochrome graphics in the DOS world, and the HP LaserJet may be well on its way to becoming a standard for laser printers. The Hercules board is supplied with a utility program (referred to as HPRINT in the Hercules documentation, but actually a part of the program HGC.COM) that replaces the DOS print screen interrupt routine and permits screen dumps of text or graphics screens to an IBM/Epson graphics printer. When it dumps to a LaserJet, however, HPRINT delivers multiple pages of garbled output because its format is incompatible.

The program described here presents the same user interface as the Hercules-supplied HPRINT utility, so existing applications that called the screen dump routine will continue to function without modification. The HPRINT screen dump is invoked by simultaneously pressing the Shift and PrtSc keys, followed by the space, 0, or 1 key, for dumps, respectively, of text, graphics page zero, or graphics page one. Alternatively, the routine may be invoked under program control by placing a space, 0, or 1 in the keyboard buffer and issuing an interrupt 5. Both text and graphics screen dumps can be interrupted at any time by pressing the Esc key.

The utility was written to be as fast as possible because, with laser printers, the picture cannot be observed as it is being printed, making long waits frustrating. The utility is memory-resident, and runs on any DOS system, since it uses only the 8088 instruction set.

The HP LaserJet Printer

The Hewlett-Packard LaserJet prints high-quality text or graphics at a resolution of up to 300 dots per inch (dpi). Printing text is conceptually straightforward. The printer accepts standard ASCII codes for printable characters. Of course, the printer also is capable of some very fancy text enhancements, including sideways printing (landscape mode), multiple fonts, multiple point sizes and character pitches, proportional spacing, and more. For the purposes of the screen dump, we do not utilize any of these features. (If an application requires that a screen dump be printed sideways in bold underlined italics, the required escape code sequences are clearly documented in the HP manuals.)

Graphics printing is central to this project, and the implementation of raster graphics on the LaserJet is also

direct. A control code sequence is sent to the printer to put it into graphics mode at one of four possible resolutions: 300, 150, 100, or 75 dpi. The resolution determines the printed pixel size. The smallest dot that can be formed by the printer has a diameter of $1/300$ inch, which forms the pixel at 300 dpi resolution. Each pixel at 75 dpi covers 16 times the paper area of a pixel printed at 300 dpi; that is, each 75-dpi pixel may be thought of as a 4-by-4 array of adjacent 300-dpi dots. Similarly, each 100-dpi pixel can be envisioned as a 3-by-3 array of dots, while each 150-dpi pixel is a 2-by-2 array. So an image at 75-dpi resolution will be 4 times larger in the x and y dimensions than the same image at 300-dpi resolution, and will be 16 times grainier. The control code to select graphics resolution is *Esc***t*#*R*, where # is replaced by the ASCII characters 75, 100, 150, or 300, depending upon the resolution desired.

After specifying the graphics resolution, another escape sequence is sent to the printer to tell it to expect raster graphics data to follow and to indicate where the graphics starting position will be. The code is *Esc***r*#*A*, where # is replaced by ASCII 0 or 1. A value of 0 indicates that the image should begin at the leftmost printable location on the page; a value of 1 indicates that the current cursor column position should be used as the left edge of the image. The concept of a printer cursor is useful in thinking about how to control the placement of printer output. Since the entire page image is constructed in the printer before the page is actually printed, you can treat the printer as if it were a video screen. In this way, programming for output to the LaserJet is much more flexible than for a line printer.

After telling the printer that raster graphics data will follow, you must tell it how many bytes of data to expect. The code is *Esc***b*#*W*, where # is replaced by the ASCII representation of the number of bytes (in decimal). The format for the graphics data itself is a simple bit-mapped representation: Each byte maps information for eight horizontally contiguous pixels, with a bit value of 1 indicating that the corresponding pixel (at the previously specified resolution) should be black. When the specified number of bytes have been received by the printer, the printer cursor is positioned at the left edge of the image area at the point where the next row of graphics will be printed, analogous to a *cr/lf* in text mode. After all the graphics have been sent, the printer is put back into text mode by the sequence *Esc***r**B*.

Text and Graphics on the Hercules Board

Text is stored in the usual manner on the Hercules board: Each character position on the screen corresponds to two consecutive memory locations, with the upper-left screen corner beginning at *B000:0000h*. The first byte contains the ASCII code for the character, and the next byte defines the character's video attributes. In this version of the screen dump, the attribute bytes are ignored for the sake of speed and compactness. There is also only a minimal translation of the IBM line graphics symbols to printer graphics. (Actually, all ASCII codes greater than 127 are converted to 127.) The result is that special characters print as a shaded rectangle on the LaserJet. Since the most common use of the upper 127 ASCII codes is to draw boxes around windows, this substitution produces printouts that often retain the character of the actual screen image without requiring the speed and code size overhead needed to translate the entire extended character set ASCII codes to the appropriate ASCII codes or graphic sequences in order to print the characters on the LaserJet. However, if you use many of these symbols and need to be able to see

them exactly on a screen dump, the implementation would be relatively simple.

The text dump is simply performed by starting at an offset of zero in segment *B000*, sending the ASCII value stored there to the printer, incrementing a counter by 2, sending the next ASCII value, and so on until 80 characters have been sent. A carriage return / line feed sequence is sent to the printer, and the process is repeated until 25 rows of data have been sent. This section should work without modification on any printer capable of recognizing ASCII codes.

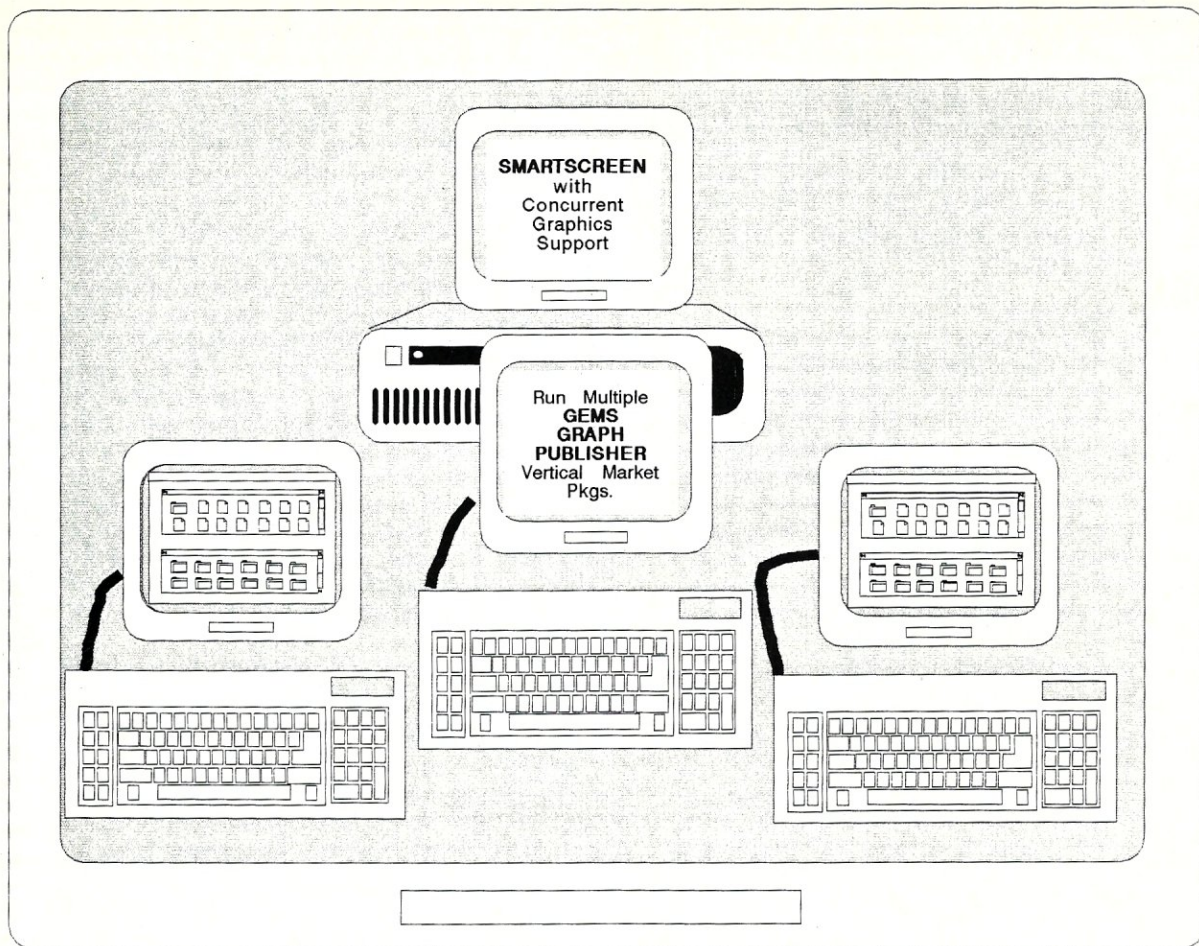
The layout of the bit image of the monochrome graphics in memory is not as straightforward as one might expect. The offset of the byte in graphics memory coding for a pixel at point (x,y) on the screen is given by the following equation from the Hercules manual: $\text{byte address} = [2000h * (y \bmod 4)] + [90 * \text{int}(y/4)] + \text{int}(x/8)$. If we want to find the location of the byte containing a specific pixel, this equation is exactly what we need. For the purposes of the screen dump, however, it is not necessary to calculate the byte address for each pixel individually. Examination of the equation shows that for each row, the beginning offset of which is described by the terms in y , all the points are stored contiguously. That is, all of the terms in x follow one another, with 8 pixels mapped per byte. Since the image is 720 pixels across, each row will be encoded in 90 bytes. So, to print a row of graphics data, we need only calculate the offset in memory for the beginning of the row, and then print the next 90 bytes of memory as raster graphics data. We then repeat this for each of the 348 rows. The only problem is to calculate the beginning offset for each row in memory.

Inspection of the above equation shows that there are two components to the offset. One component is composed of $2000h * (y \bmod 4)$. This means that row 1 will be located $2000h$ bytes higher in memory than row 0, row 2 will be $2000h$ bytes higher than row 1, and so on. The next component, $90 * \text{int}(y/4)$, simply provides a 90-byte step function through each of the four $2000h$ byte areas. Thus row 4 will be the next 90 bytes following the 90 bytes that constitute row 0, and will be in the area with an overall offset of 0. Similarly, row 5 will be the 90 bytes following row 1, and will be in the area that began at $2000h$. This is perhaps not the most simply organized scheme, but it should be apparent by now that it is not necessary to perform any multiplications, divisions, or logical ANDs ($[y \bmod 4]$ is the same as $[y \text{ and } 3]$), in order to calculate the beginning offset of each successive row of graphics data. The procedure is as follows:

1. Set an offset counter to 0.
2. Print the row of 90 bytes of graphics data beginning at the offset.
3. Add $2000h$ to the offset counter. Repeat step 2.
4. Repeat step 3.
5. Repeat step 3 again. (We have now printed four rows of graphics. The next row is back in area zero.)
6. Subtract $6000h$ from the offset counter, to point back to the first area, then add 90 to point past the row we just printed. This is the same as subtracting $5FA6h$.
7. Loop back to step 2 until we have printed 348 rows—that is, 87 loops through the routine.

This approach is compact, easy to code, and very fast. Similar strategies can be developed for accessing interleaved data for other types of display adapters.

The Hercules board memory supports two pages of graphics data. Page 0 begins at *B000:0000h*, while page 1 begins at *B800:0000h*. Different application programs



SMARTSCREEN

CONCURRENT SUPPORT FOR MULTIUSER GRAPHICS WORKSTATIONS

A multi-user DOS operating system with full graphics support!

Smartscreen uses **cost-efficient** monitors and keyboards for 1-8 users on 80386-based computers with monitor multiplexer boards to create a **high-performance** multiuser system with **graphics** capabilities for **every user**.

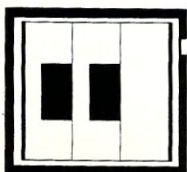
Smartscreen, an enhanced version of Concurrent Dos 386 Release 2.0, supports **full Hercules graphics** or **Color/EGA capabilities** and **multitasking** at every station, and runs PC-DOS and Concurrent software for maximum flexibility. You can have **two tasks** at every monitor for increased productivity. **Smartscreen** also supports local devices - mouse, serial printer and optional parallel printer - at each station for added flexibility.

Smartscreen allows application software to write directly to the video RAM at all stations for high performance and full PC graphics capabilities, eliminating screen emulation and serial slowdown.

Smartscreen has special versions available for optimized performance of selected vertical market packages, including MC Software's **INCOME & INMASS**, **Medical Manager**, **Digital Dining**, **Dataflex** and other **DBMS**-based applications. Also supported are most **desktop publishing** packages and selected **design and CAD** packages.

SMARTSCREEN includes **CCI support** - prompt, professional support when you need it, tailored to dealers, distributors, and developers. **For more details on Smartscreen and supported multiplexer boards**, call us today.

Smartscreen is a trademark of Concurrent Controls, Inc., Concurrent is a trademark of Digital Research, Inc.



CONCURRENT CONTROLS, INC
A SOFTWARE ENGINEERING COMPANY

3770 24th Street, Suite 206 San Francisco CA 94114 (415) 648-2174 FAX (415) 648-0340

CIRCLE 53 ON READER SERVICE CARD

use different pages for their graphics. Lotus 1-2-3 uses page 1, while HBASIC, Hercules' modified version of BASICA, uses page 0. To print page 0 graphics, set the ES register to *B000h* and proceed as described to calculate the offset into this segment for each row of graphics data. For page 1, set ES to *B800h* instead. Note that page 0 graphics data occupies the same address space that text does on the Hercules board. Thus a graphics dump of page 0 while the board is in text mode will not produce meaningful output, although a dump of graphics page 1 will work fine.

Aspect Ratio Correction

The aspect ratio of a graphics image is the ratio of its width to its height. On the LaserJet, each pixel produced is as wide as it is tall, but a pixel on the screen is taller than it is wide. Thus if we draw a box with the same number of pixels in the *x* axis as in the *y* axis, on the printer it will appear to be a square, but on the screen it will appear as a rectangle that is taller than it is wide. Similarly, an image that appears to be a circle on the screen will be seen as a flattened oval on the printer. To have the printed screen dump image appear the same as the screen image, it is necessary to perform an aspect ratio correction.

The Hercules manual states that the aspect ratio of the screen graphics image is 2:3; that is, each pixel can be thought of as an array of two equally spaced columns of three equally spaced dots each. (A pixel is really the smallest addressable graphics unit; the concept of a 2-by-3 array is merely a useful fiction.) With the LaserJet in the 150-dpi mode, each pixel is made up of a 2-by-2-square matrix of dots. So aspect ratio correction can be accomplished simply by printing a row of raster graphics data, using printer control commands to move the laser cursor back up one dot, and then reprinting the row of data. The top line of dots of the second row is superimposed on the bottom line of dots of the top row, yielding the 2-by-3 pixel which we sought. The command to move the laser cursor is *Esc *p#Y*, where # is the ASCII decimal representation of the vertical position in terms of dots. If the number is prefaced by a - or + symbol, the number is treated as an offset relative to the current cursor position. Thus, to send the cursor up one dot, the control sequence is *Esc *p-1Y*. Only the LaserJet Plus and Series II support cursor positioning in terms of dots, although the regular LaserJet does support positioning in terms of decipoints (1 decipoint = $1/720$ inch).

The actual aspect ratio of the screen image depends to some extent on the manufacturer and adjustment of the video monitor you use, so the aspect ratio correction may not be perfect for your system. However, for most purposes the correction will produce a printed image that will be close enough.

Sending the Data to the Printer

DOS provides us with several possible methods for sending data to the printer. The first option is to use the printer support functions available via interrupt *21h*. Function 5 sends 1 byte of data to the current list device, normally LPT1 unless redirected. This function will provide the highest possible degree of compatibility between various systems, but is very slow. An alternative is to use the interrupt *17h* BIOS printer support functions. These routines also send data to the printer 1 byte at a time, but they are faster than the higher-level interrupt *21h* functions by about a factor of 6.

The conventional wisdom has been to avoid using BIOS-level routines directly because printers in general are too slow to accept data much faster than interrupt

21-level routines can drive them, and incompatibility between systems could become a factor. The LaserJet is more than fast enough to keep ahead of the interrupt 21-level printer support, at least on a 4.77-MHz, 8088 machine. The major incompatibility between systems that you will encounter in using the BIOS routines is in the logical installation of the printer. The interrupt 21h function 5 always prints to the current list device, while the BIOS routines require that you specify which parallel adapter you are sending the data to. Thus, the screen dump assembled to print data on LPT1 will not function properly on a system that has the LaserJet hooked up as LPT2. In the case of this utility, since the parallel port built into the Hercules board is defined as LPT1, it seems natural to assume that the LaserJet will be connected here. If it is not, changing a single equate at the beginning of the program and reassembling will correct the problem.

Since the LaserJet Plus comes with both parallel and serial interfaces, some people may be using their printers on the serial instead of parallel interface. In this case, the serial transmission will be the rate-limiting factor for the screen dump. The utility can easily be rewritten to handle a serially interfaced printer, but it will be relatively slow. If speed is at all important to your application, use the parallel interface.

It is also possible to double the speed by accessing the printer port directly. Each parallel adapter is controlled by three I/O registers accessible through three consecutive microprocessor port addresses. The lowest of the port addresses is called the base address for the adapter. The register accessed through the base address port is the data register. As the name implies, the data to be sent is placed in this register. The register at port base address +1 is the status register. Bit 7 of this register is set to 1 if the printer is not busy, and 0 otherwise. The third register is the control register, accessed via the port at base address +2. Setting bit 0 of this register will cause the data byte in the data register to be transmitted to the printer. Bits 2 and 3 of the control register should normally be set, so outputting a value of 13 (decimal), followed by 12, will send the byte in the data register and then return the control register to the normal state.

The BIOS keeps track of where the ports for each adapter are located in an area of memory beginning at *0040:0008h* for LPT1, *0040:000Ah* for LPT2, and *0040:000Ch* for LPT3. The addresses stored at these locations are those of the base address of each adapter. If an adapter is not present, a value of 0 is stored. The storage of the port addresses in low memory by DOS also provides an opportunity to write a simple program for swapping the base address values for two of the adapters, thus obviating the need for reassembling the screen print program when changing printer ports.

To access a parallel adapter directly, it is only necessary to read the base address from low memory, output the data byte to be sent to the port, toggle bit 0 of the port at base address +2 momentarily to 1 to transmit the data, then monitor bit 7 of the status register port at base address +1 until the printer is ready to accept another byte.

The DOS Print Screen Flag

DOS maintains a flag byte at *0050:0000h* that tells a screen print routine whether or not a screen print is permitted. To maintain compatibility, a screen print routine must check this flag upon entry. If this byte has a value of 1, exit immediately without printing. If the value is 0, set the flag to 1 upon entry, so that a second screen print cannot interrupt the one currently in progress. When finished, reset the flag to 0.

Writing Memory-Resident Programs

The first step in making a program memory-resident is to write it as a stand-alone routine. Once the routine is debugged, the conversion to memory residency begins by adding code that will preserve the machine environment when the program is invoked. This is necessary because the program may be invoked while some other program is executing, and there is no way to know *a priori* what registers may be in use. The solution is simply to preserve all registers used by the routine immediately upon entry, and to restore them immediately prior to exiting. This includes registers that may be altered by DOS during the performance of various system calls, even if you do not explicitly use these registers yourself.

The next modification for a memory-resident interrupt handler is to enable interrupts as soon after entry as is feasible with an *sti* instruction. This is necessary because whenever a software interrupt is invoked, the interrupt flag is automatically set, preventing hardware interrupts from occurring.

Finally, the routine must be modified to exit via an *iret* instruction, instead of via the normal exit routine employed (*int 20h*; *int 21h*, function 0; or *int 21h*, function 4ch).

To summarize, write and debug the routine first, then add a code section to store all the registers at the beginning and to enable interrupts, and change the termination procedure to a simple *iret* instruction immediately preceded by a restoration of the initial register contents. The only thing that remains to be done is to write an initialization routine that actually does the interrupt replacement.

The steps that the initialization code must perform are to patch the interrupt jump vector in low memory with the address of the new routine, and to terminate but leave the interrupt processing portion of the code resident. Setting the new interrupt address in low memory is easily accomplished by using *int 21h*, function 25h. Simply place the segment:offset of the new routine in *ds:dx*, and the number of the interrupt to be replaced in *al* prior to calling this function. Do not write directly to low memory yourself if you want to increase the likelihood that your interrupt handler will function under various multitasking systems, which may have multiple copies of the interrupt jump vector stored at various locations.

DOS provides two mechanisms by which programs are able to terminate but remain resident in memory. The first and perhaps more common method is to use the interrupt 27h DOS function. The other method, employed here, is to use function 31h of interrupt 21h. This function expects the length of the resident code (in paragraphs) in *dx*, the function number (31h) in *ah*, and an optional program termination code in *al*. Note that the paragraph count expected in *DX* must include the 100h-byte length of the program segment prefix, so increase the number of paragraphs calculated for your code by 16 or you will end up with only part of your routine memory-resident. The advantage to using the function 31h mechanism for making a routine resident instead of using the *int 27h* method is that the latter is limited to programs smaller than 64K in length, and is really designed to work with .COM files instead of .EXE files.

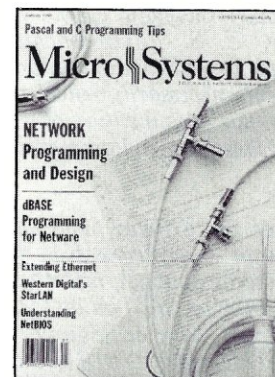
It is also desirable to have the initialization code determine whether or not the routine has been previously loaded to avoid progressively filling up memory with unnecessary multiple copies. No standardized method for flagging the installation of user-installed memory-resident programs exists, but at least three methods have come into common use. The first is to set a byte in memory to a flag value upon installation of the routine.

Tell Us What You Want - And Get What You Need

Now you can get the product and service information you need with reader service, thanks to *Micro/Systems Journal*.

By filling out the reader service card, you receive the information you need from our advertisers. This includes facts on specific products or additional literature on their services. *And* when you answer the short editorial questions, you tell us what you want to read about in future issues. Finally, by responding to our advertisers through reader service, you contribute to the growth and success of *Micro/Systems*.

To get your free information, look for the reader service numbers listed at the end of articles and ads. Then simply fill in your name and address on the card opposite page 40, answer the editorial questions, circle the number(s) of your choice, and drop the card in the mail. It's that simple. And it's absolutely free.



The problem with this approach is that you must be certain that other programs will not corrupt this location. Another often-used technique is to take over an unused interrupt vector and use this location to store a flag value. This suffers from the same problem as the first method, in that it is impossible to know if another program will also make use of the "unused" vector.

The method I have used does not depend on the value of a flag byte, but rather on the structure of the program itself. Interrupt 21h, function 35h, returns the offset of the first instruction of the requested interrupt handler in the bx register. If the first executable instruction is not at the beginning of the memory-resident code, the offset will be nonzero. If we store program data in memory prior to storing executable code, we can be assured of having a nonzero offset that, since it depends on the amount of local data storage used by our program, has a high likelihood of being unique. The initialization routine simply requests the current offset of the interrupt to be replaced from DOS and compares the value returned to the offset of the first executable statement of the process to be installed. If the values agree, then it is likely that the routine is already present.

Conclusion

The routine in Listing 1 is a simple implementation of the features described here. The program assembles with Microsoft's MASM, Version 4.0, and has been tested on a Leading Edge Model D using an NEC V20 microprocessor at 4.77 MHz, using both the built-in Hercules emulation and a Hercules board, as well as on an IBM PC/XT, an IBM PC/AT, and an AT running a 16-MHz, 80386 board, all using Hercules boards. The program works equally well on both the LaserJet+ and LaserJet Series II printers.

Attempts have been made to keep the execution speed high and the code size minimal. Interestingly, the program runs as fast on an 8088 machine as it does on the 80386, indicating that the speed at which the printer can accept data is the rate-limiting factor. It should be relatively simple to adapt the techniques used in this program to other video adapters and other printers, as well as to add various enhancements, such as reproducing underlining and blanking on text dumps or landscape-mode graphics screen prints at different resolutions. □

Dr. Jeffrey Ross is a research biologist in the Genetic Toxicology Division, Carcinogenesis and Metabolism Branch of the U.S. Environmental Protection Agency. He is interested in the use of computers for scientific data analysis.

This article has been reviewed by the Health Effects Research Laboratory, U. S. Environmental Protection Agency, and approved for publication. Approval does not signify that the contents necessarily reflect the views and policies of the Agency, nor does mention of trade names or commercial products constitute endorsement or recommendation for use.

All the source code for articles published in Micro/Systems is available on an MS-DOS disk. To order, send \$14.95 to Micro/Systems Journal, 501 Galveston Drive, Redwood City, CA 94063; or call Tim at (415) 366-3600. Please specify the issue number. Source code is also available on CompuServe; type GO DDJ and check the data library, line 14.

Did you find this article particularly useful?
Circle number 5 on the reader service card.

Advertiser Index

RSN	Advertiser	Page
	Aker Corporation	28
78	Andsor Research Inc.	19
62	Austin Code Works	61
58	CAE/SAR Systems, Inc.	15
96	Command Computer Corporation	55
53	Concurrent Controls	37
64	Controlled Printout Devices	52
	DBMS	57
91	Data Tech Institute	33
	Digital Research Computers	43
98	Digital Research, Inc.	13
42	Ecosoft Inc.	45
43	Essential Software	10
75	Gimple Software	27
105	Hauptpage Computer Works	C-2
54	IGC	7
	Invisible Software	65
60	Magna Carta Software	55
97	Max Software Consultants	69
55	MetaWare Incorporated	11
56	MetaWare Incorporated	33
	Micro/Systems Journal	32,53
	Nanosoft Associates	67
65	Novell Development Division	2
103	Nu-Mega Technologies	34
46	PC Tech	30
66	Periscope Company, Inc.	66
79	Personal Tex, Inc.	62
	Programmer's Power Pack	53
82	QNE International	9
67	Qualstar Corporation	52
86	Quarterdeck Office Systems	4-5
90	Quarterdeck Office Systems	C-3
95	Rainbow Technologies	72
94	Santa Cruz Operation (The)	C-4
47	Semi-Disk Systems	63
	Slicer Computer	30
	Softfocus	54
81	Stargate Technologies, Inc.	47
73	Sumware, Inc.	18
80	Teletex Enterprises, Inc.	16
	Vermont Creative Software	1
92	Whitesmiths' Ltd.	71
59	Wyte	69

Listing 1. PSCR . . . Print SCREEN Utility

This print screen program emulates the HPRINT program for the Hercules monochrome graphics board and operates with the HP LaserJet+ printer. (Jeff Ross, 1987)

```

;*****
;***** PSCR.....Print SCREEN utility for *****
;***** Hercules monochrome graphics board *****
;***** and HP LaserJet+ printer. *****
;***** Emulates the Hercules HPRINT *****
;***** program. *****
;***** Jeff Ross, 1987 *****
;*****
cseg segment
assume cs:cseg,ds:cseg,es:nothing,ss:nothing
;
;set following equate for parallel adaptor actually used
printer equ 0 ;parallel adaptor # (0-lpt1, 1-lpt2, 2-lpt3)
;
main proc far
    jmp install
;
;*****
;*** Data Storage Area
;*****
out_data_port:
    dw (?)
;local storage for base port address for parallel adaptor
init:
    db 1BH,"p200X",1bh,"t150R",1bh,"r1A"
;put printer in raster graphics mode, left margin at 2/3",
;150 dpi resolution
reset:
    db 1bh,"xB",1bh,"E"
;graphics finished, back to normal mode,eject page
;
rowout:
    db 1bh,"b90W" ;90 bytes of graphics data to follow
;
upline:
    db 1bh,"p-1Y" ;laser cursor up 1 dot
;
;*****
;*** Interrupt Replacement Begins Here
;*****
execute:
    sti ;before anything else, enable interrupts
;next, save everything we use
    push es
    push bx
    push cx
    push dx
    push si
    push di
    push bp
    push ds
    push ax
start:
    call setflag ;set flag, and check to see if a screen print
                ;was already underway
    jz exit2 ;if so, don't start a new one now
    call pchk ;see if printer is functional
    jnz exit1 ;if not, just quit
;set ds to point into cseg
    mov ax,cseg
    mov ds,ax
;
;read key value to determine which screen to print
;
get_key:
    xor ah,ah ;ah=0, read character function
    int 16h ;wait for next key from keyboard
check_key:
    cmp al,32 ;was the character a space?
    jnz not_text ;if not, do a graphics dump
;
; space key was pressed, do text dump
;
    mov di,25 ;initialize row counter
    mov ax,0b000h ;beginning of text video memory
    mov es,ax ;into es
assume es:nothing
    xor bx,bx ;bx=offset into video memory, initially 0
tx:
    mov cx,80 ;80 columns per row
esc?:
    mov ah,01 ;keyboard status function
    int 16h ;read status

```

```

    jz tx1 ;go on if no key
    xor ah,ah ;(ah=0, read char) if there was a key,
    int 16h ;get ascii value
    cmp al,1bh ;see if it was the escape key
    je exit ;if so, exit, otherwise ignore it
tx1:
    mov al,es:[bx] ;get ascii char
;
;This is where translation of special characters would occur, i.e.,
;characters with ASCII values 128-255. Here I just translate all of
;them to a rubout character.
    cmp al,127 ;is character >= 127
    jnae notupper ;if not, just print it
    mov al,127 ;else set all special characters to rubout
notupper:
    xor ah,ah ;ah=0, print char function
    mov dx,printer ;dx=parallel port #
    int 17h ;print it
    inc bx ;increment offset pointer to attribute byte
    inc bx ;and increment again to point to next ascii byte
    loop tx1 ;do it again until 80 characters are sent
    mov ax,13 ;send a cr, ah=0, al=13
    int 17h ;print it
    mov ax,10 ;and lf, ah=0, al=10
    int 17h ;print it
    dec di ;decrement row counter and
    jnz tx ;do it all again if not through with 25 rows
;
exit:
;
;Exit point if everything ok. Eject page, reset flag, and return.
;
    mov bx,offset reset ;reset printer and eject page string
    mov cx,6 ;string length
    call print
;
exit1:
;exit point if initial printer problem
;
    call resetflag ;reset the print screen flag to zero
exit2:
;exit point if screen print flag not zero, dont reset it!!
;
;restore all registers to entry values
    pop ax
    pop ds
    pop bp
    pop di
    pop si
    pop dx
    pop cx
    pop bx
    pop es
;and return from interrupt
    iret
;
not_text:
;
;al still holds ascii of key entered, if not ascii 1, do page 0
;regardless of which key was pressed
    mov bx,0b000h ;page 0 base address for graphics page 0 memory
    cmp al,49 ;see if it's page 1
    jne do_graph ;if not page 1, do page 0
    add bx,8 ;page 1 base address = b800h
;
do_graph:
    mov es,bx
;put printer into raster graphics mode
    mov cx,19 ;length of message into cx
    mov bx,offset init ;pointer to raster_mode message into bx
    call print
;
    mov si,87 ;graphics row counter, initially 87 (4*87-348)
    xor bp,bp ;starting offset into graphics space
;
mainloop:
;this is start of main loop
;
;check for escape key to interrupt screen dump
esc?:

```

... listing continues


```

mov ah,01          ;keyboard status function
int 16h           ;read status
jz next4          ;go on if no key
xor ah,ah         ;if there was a key, get ascii value
int 16h
cmp al,1bh        ;see if it was the escape key
je exit          ;if so, exit, otherwise ignore it
;no escape key, so print 4 more rows of dots
next4:
call printrow     ;print graphics data from 1st area
add bp,2000h      ;point to 2cnd area
call printrow     ;print graphics data from 2cnd area
add bp,2000h      ;point to 3rd area
call printrow     ;print graphics data from 3rd area
add bp,2000h      ;point to 4th area
call printrow     ;print graphics data from 4th area
sub bp,5fa6h      ;(same as + 90 decimal - 6000h)
dec si           ;decrement row counter
jnz mainloop      ;if not finished, do another row
jmp exit         ;if finished, quit

;
main endp
;
;***** Subroutines Follow *****
;*****
;*** printrow---prints row of raster graphics at es:bp using low-
;*** print driver. Sends each row twice for aspect ratio correction
;
;***** expects beginning offset in bx, segment in es
;leaves es, bx unchanged, consumes di,ax,bp,cx,dx
;*****
printrow proc near
mov di,2          ;loop counter for aspect ratio correction
mov dx,word ptr [out_data_port] ;base port address
lineout:
mov cx,6          ;length of transfer_raster_graphics string
mov bx,offset rowout ;pointer to string
rwout:
mov al,[bx]       ;next byte of string
out dx,al         ;put into data register
inc dx           ;point past status register
inc dx           ;to control register
mov al,13         ;set bit 0=1, (normal value is 12)
out dx,al         ;send the byte
dec al           ;return control reg. to normal setting (12)
out dx,al         ;send it
dec dx           ;point back to status register
wait1:
in al,dx          ;read status register
test al,80h       ;check bit 7
jz wait1         ;if 0, printer still busy, so keep checking
dec dx           ;byte sent, so point back to data register
inc bx           ;point to next byte of string
loop rwout        ;repeat until cx=0
;
;send the video data now
;
mov cx,90         ;cx=bytes per row
mov bx,bp         ;bp=video ram offset calculated earlier
prntgrf:
mov al,es:[bx]    ;loop to print raster data
out dx,al         ;(same procedure as above)
inc dx
inc dx
mov al,13
out dx,al
dec al
out dx,al
dec dx
wait:
in al,dx
test al,80h
jz wait
dec dx
inc bx
loop prntgrf
;
;*****Aspect Ratio Correction
;
dec di           ;is this second time printed this row?
jz next         ;if yes, do next row
mov cx,6         ;else, move laser cursor up 1 dot and
mov bx,offset upline ; and send again
upline:
mov al,[bx]      ;same procedure as before
out dx,al
inc dx
inc dx

```

```

mov al,13
out dx,al
dec ax
out dx,al
dec dx
wait2:
in al,dx
test al,80h
jz wait2
dec dx
inc bx
loop upline
jmp lineout      ;send line again
next:
ret
printrow endp
;*****
;print subroutine to output ascii data using bios int 17h
;on entry: cx=number of bytes of ascii to output
;         bx=pointer to location of start of string
print proc near
prn: mov al,[bx]
xor ah,ah         ;ah=0, print char function
mov dx,printer    ;dx=parallel adaptor #
int 17h
inc bx
loop prn
ret
print endp
;*****
;printer status check
;returns with zero flag set if printer ok
;
pchk proc near
mov dx,printer ;parallel adaptor #
mov ah,02       ;printer status function
int 17h
test ah,29h     ;returns z if no error, nz if error
ret
pchk endp
;
;*****
;DOS uses byte at 0050:0000h to indicate whether or not it is ok to
;do screen print, 0=ok, 1=don't print. To prevent one screen print
;from interrupting another, set flag upon entry, reset to 0 on exit
;
;setflag tests flag, sets it to 1, and returns with zero set if
;screen print flag was already 1.
;
;setflag just sets the flag to 0.
;
setflag proc near
mov ax,0050h
mov es,ax        ;es=0050
assume es:nothing
xor bx,bx        ;bx=0
mov al,01
cmp es:[bx],al   ;set zero flag if already set
mov es:[bx],al   ;set flag to 1
ret
setflag endp
;
;
resetflag proc near
mov bx,0050h
mov es,bx        ;es=0050
assume es:nothing
xor bx,bx        ;bx=0
mov byte ptr es:[bx],bl ;0050:0000=0
ret
resetflag endp
;*****
pgm_length equ $-main ;length of resident part of code in bytes
;*****
;***** Beginning of Install Routine *****
;*****
;as much of operations of program as possible should be done here
;e.g., set up printer port and store data not needed by resident
;portion, so that resident code size can be kept small.
;
install proc near
mov ax,cseg
mov ds,ax        ;point ds into cseg
assume ds:cseg
mov ax,3505h     ;ah=35h (get interrupt vector),al=05 (interrupt #)
int 21h          ;get vector
cmp bx,offset execute ;if same as our offset, it's probably us
je already_here   ;so don't re-install
;

```



```

;set up parallel port # for print routines...remember-set equate
;at beginning
    mov ax,40h          ;get port number for lpt1:
    mov es,ax           ;for use in low-level print driver
assume es:nothing
;get base port address for parallel adaptor # printer and store locally
    mov cx,es:[8+(printer*2)]
    mov word ptr [out_data_port],cx
;
;replace vector to point to us
    mov ax,2505h        ;ah=25h (set interrupt vector),al=5 (interrupt #)
    mov dx,offset execute ;dx=offset for replacement vector execution start
    int 21h             ;change int 5 vector to point to our routine
;
    mov dx,offset installed
    mov ah,09           ;tell world we're installed
    int 21h
;calculate # of paragraphs of code + psp to remain resident
    mov dx,((offset pgm_length+15)/16)+16
;dx now has the number of paragraphs to save
    mov ax,3100h
    int 21h             ;terminate, but keep processor code resident
;
;if we are already installed...
already_here:
    mov dx,offset already_here_msg
    mov ah,09           ;tell them that we're already here
    int 21h
    mov ah,04ch         ;and then just exit back to DOS
    int 21h
;
;local data storage
already_here_msg:
    db "Print SCreen for HP LaserJet+ already resident",7,10,13,"$"
installed:
    db "Print SCreen for HP LaserJet+ now resident",10,13,"$"
install endp
cseg ends
;
end

```

End Listing 1

THE NEW 65/9028 VT ANSI VIDEO TERMINAL BOARD!

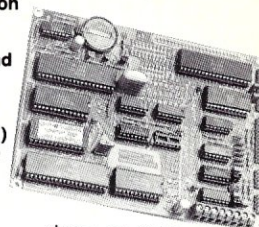
★ FROM LINGER ENTERPRISES ★

A second generation, low cost, high performance, mini sized, single board for making your own RS232 Video Terminal. This highly versatile board can be used as a stand alone video terminal, or without a keyboard, as a video console. VT100, VT52 Compatible.

FEATURES:

MICRO SIZE!

- ★ Uses the new CRT9128 Video Controller driven by a 6502A CPU
- ★ On-Screen Non-Volatile Configuration
- ★ 10 Terminal Modes: ANSI, H19, ADM-5, WYSE 50, TVI-920, KT-7, HAZ-1500, ADDS 60, QUME-101, and Datapoint 8200
- ★ Supports IBM PC/XT, and Parallel ASCII Keyboards
- ★ Supports standard 15.75 kHz (Horiz.)
- ★ Composite or Split Video (50/60 Hz)
- ★ 25 X 80 Format with Non-Scrolling User Row
- ★ Jump or Smooth Scroll
- ★ RS-232 at 16 Baud Rates from 50 to 19,200
- ★ On Board Printer Port
- ★ Wide and Thin Line Graphics
- ★ Normal and Reverse Screen Attributes
- ★ Cumulative Character Attributes: De-Inten, Reverse, Underline and Blank
- ★ 10 Programmable Function Keys and Answerback message
- ★ 5 X 8 Character Matrix or 7 X 9 for IBM Monitors
- ★ Mini Size: 6.5 X 5 inches
- ★ Low Power: 5VDC @ .7A, ± 12VDC @ 20mA.



\$79⁹⁵

FULL KIT

w/100 Page Manual

ADD \$40 FOR A&T

OPTIONAL EPROM FOR PC/XT STYLE SERIAL KEYBOARD: \$15

SOURCE DISKETTE: PC/XT FORMAT 5 1/4 IN. \$15

Digital Research Computers

P.O. BOX 381450 • DUNCANVILLE, TX 75138 • (214) 225-2309

Call or write for a free catalog on Z-80 or 6809 Single Board Computers, SS-50 Boards, and other S-100 products.

TERMS: Add \$3.00 postage. We pay balance. Orders under \$15 add 75¢ handling. No C.O.D. We accept Visa and MasterCard. Texas Res. add 6-1/4% Tax. Foreign orders (except Canada) add 20% P & H. Orders over \$50 add 85¢ for insurance.

CANON 80 COLUMN PRINTER - \$29.95

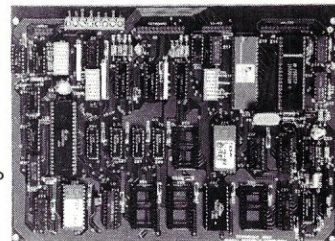
ORIGINALLY MANUFACTURED FOR THE PC JR. BUT WITH OPTIONAL CONNECTOR WILL WORK WITH PC, XT, OR AT. REQUIRES SERIAL I/O. THIS THERMAL PRINTER IS QUIET AND USES EASY TO GET 8 1/2 IN. ROLLS OF PAPER, 50 C.P.S., UPPER AND LOWER CASE, PLUS GRAPHICS. ORIGINAL LIST PRICE \$199.00. ADD \$3.00 FOR PC/XT CONNECTOR. ADD \$5.00 UPS.

THE NEW ZRT-80 CRT TERMINAL BOARD!

A LOW COST Z-80 BASED SINGLE BOARD THAT ONLY NEEDS AN ASCII KEYBOARD, POWER SUPPLY, AND VIDEO MONITOR TO MAKE A COMPLETE CRT TERMINAL. USE AS A COMPUTER CONSOLE, OR WITH A MODEM FOR USE WITH ANY OF THE PHONE-LINE COMPUTER SERVICES.

FEATURES:

- ★ Uses a Z80A and 6845 CRT Controller for powerful video capabilities.
- ★ RS232 at 16 BAUD Rates from 75 to 19,200.
- ★ 24 x 80 standard format (60 Hz).
- ★ Optional formats from 24 x 80 (50 Hz) to 64 lines x 96 characters (60 Hz).
- ★ Higher density formats require up to 3 additional 2K x 8 6116 RAMS.
- ★ Uses N.S. INS 8250 BAUD Rate Gen. and USART combo IC.
- ★ 3 Terminal Emulation Modes which are Dip Switch selectable. These include the LSI-ADM3A, the Heath H-19, and the Beehive.
- ★ Composite or Split Video.
- ★ Any polarity of video or sync.
- ★ Inverse Video Capability.
- ★ Small Size: 6.5 x 9 inches.
- ★ Upper & lower case with descenders.
- ★ 7 x 9 Character Matrix.
- ★ Requires Par. ASCII keyboard.



\$89⁹⁵

A&T

ADD

#ZRT-80 \$50

(COMPLETE KIT, 2K VIDEO RAM)

**OUR BEST
SELLER!**

FOR 8 IN. SOURCE DISK
OR PC-XT FORMAT 5 1/4 IN.
ADD \$10

Digital Research Computers

P.O. BOX 381450 • DUNCANVILLE, TX 75138 • (214) 225-2309

Call or write for a free catalog on Z-80 or 6809 Single Board Computers, SS-50 Boards, and other S-100 products.

TERMS: Add \$3.00 postage. We pay balance. Orders under \$15 add 75¢ handling. No C.O.D. We accept Visa and MasterCard. Texas Res. add 6-1/4% Tax. Foreign orders (except Canada) add 20% P & H. Orders over \$50 add 85¢ for insurance.

CALL ADVERTISER DIRECTLY

Actor

*This language simplifies writing applications
for Microsoft Windows.*

by Michael Guttman

Until recently, software developers wishing to interface their applications with Microsoft Windows had only one choice, the Microsoft Windows Software Development Kit (the SDK). In February 1987, however, the Whitewater Group introduced Actor, an interactive, object-oriented programming language specifically designed for the Windows environment.

Actor offers several features that are not available in the SDK:

1. A complete, interactive programming environment based around Windows that includes its own editor, compiler, and debugger.
2. Automatic handling of Windows messages, calls, and memory management, which frees the user from needing to know or consider most of the technical details of interfacing to Windows.
3. A substantial library of object-oriented programming routines (supplied in nearly all cases with source code) that can be readily tapped to create new applications.

The main attraction of these features is that a programmer new to Windows should be able to create his first Windows application using Actor within days, even if that programmer's previous programming experience is with a language other than C or assembler. By contrast, typical ramp-up time under the SDK can easily be several months, even *assuming* considerable experience with C or assembler.

My associates and I tested how well Actor performed using Version 1.1 running under both Windows 1.04 and 2.03 on ordinary 8-MHz, 640K, IBM PC/AT with a 20-MB disk

drive. This review reflects about 150 hours of experience with Actor over a six-week period. We collectively had many years of higher-level language and database programming, but absolutely no previous experience with Windows programming.

The Actor Environment

Actor is supplied with a manual and six diskettes, four of which contain a run-time version of Windows. If you already own a copy of Windows, this run-time version can be dispensed with, and you should be able to upgrade to new Windows versions at any time without affecting Actor.

The two other diskettes contain the ACTOREXE file, various Actor *image* (.IMA) files, a set of Actor *class* (.CLS) files, and various utility files and programs. An INSTALL program automatically installs Actor in about five minutes on the subdirectory of your choice.

Once installed, Actor runs like any other Windows application. For example, the DOS command WIN ACTOREXE ACTOR.IMA will load the Actor program, which will in turn load ACTOR.IMA, an image of the Actor environment, normally the last one saved by the user.

If enough memory is available, other Windows applications can be run concurrently with Actor. On a 640K machine and a full-blown version of Actor, however, we didn't have enough extra room for more than the Microsoft Windows Executive and a few very small programs, such as Microsoft's Clock.

Actor begins by creating a tiled Window called Actor Display. Much like a system console, Actor Display is the output area for Actor's error and status messages, although these can easily be directed elsewhere if desired. Actor Display also accepts commands, but its editing capabilities

are somewhat limited.

The Actor initialization sequence also creates a more interactive pop-up window called Actor Workspace. Actor Workspace is graced with a menu bar and a full-featured text-editing area primarily intended for entering Actor commands.

Workspace is also the jumping-off area for *Browser*, another editing window that can be invoked from the Workspace menu bar. As its name implies, Browser allows you to browse through (and modify) the entire body of Actor code, including Browser itself. Virtually all programming is normally done in Browser. Multiple concurrent sessions of Browser can be invoked from Workspace, a handy feature both while developing and debugging.

Workspace also includes a menu bar option for *Inspector*, another specialized workspace for examining the contents and structure of Actor objects. Other Workspace menu options are a general-purpose text file editor, a *Snapshot* utility for creating new *image* files, a *Cleanup* utility for static area garbage collection, and a *Dolt* utility for executing interactive commands.

The Actor Language

Actor is an object-oriented language built around a Forth-like threaded interpreter, which contains a number of specialized features for interacting with Microsoft Windows. At the very core of Actor is a kernel that handles all of the actual code interpretation and execution, and a set of code primitives that handles low-level functions. An ordinary Actor user has no access to the kernel or the primitives.

Above the Actor kernel and primitives is a set of language implementation routines, written in Actor, that handle parsing, interpretation, and

compilation of Actor code. The source code for the most general of these routines is available to the user and can be quite useful; source code for routines proprietary to Actor itself is not supplied.

Above the language routines are the Actor tools, including Workspace, Browser, and Inspector. Nearly all the source code for these tools are provided, and they serve as fairly good examples of how an Actor/Windows application is implemented. Of course, if you don't like any of these tools, they can be modified, as can any other Actor application. Furthermore, the Actor tools can be discarded in a "sealing off" process once a user application is completed and ready to stand alone.

All Actor code is divided into classes, the source of which typically resides in the .CLS files mentioned earlier. A *class* is a collection of objects that share the same behavior; the behavior of a given class is defined as a set of *methods*, written in Actor code. In a sense, methods are very similar to subroutines in a traditional programming language, and Actor code is not far different from C or Pascal.

Every object of a certain class is referred to as an *instance* of that class. When created, each object is immediately endowed with a set of *instance variables* that are used to keep track of its current state. While each object has its own unique set of instance variables, these variables have the same name for every object of a given class.

If two objects of the Window class are created, for example, each will have an instance variable called *locRect*, which defines the current position of its window on the screen. Naturally, however, the value in *locRect* would be different for each object, unless they happen to overlap precisely.

The classes of Actor are arranged in a hierarchy in which each class has an ancestor from which it inherits its initial characteristics. When a new class is created, its objects inherit the instance variables and methods of every one of its ancestor classes, unless these have been redefined for the descendant class.

For this reason, many Actor classes are often *formal* classes that describe general behavior destined to be inherited by their descendants. For example, the Window class would not normally be used to actually generate objects (although, as we show below, it can be); the more likely use of the Window class is as the ancestor of other Window classes, each with its

own specialized behavior.

An excellent example of the topology of object-oriented programming is laid out in Browser (see Figure 1). Browser is a window with four child windows. The upper-left window shows a portion of the hierarchy of classes through which the user can scroll and select a desired class to "browse."

Once a class is selected, the upper-middle window shows a list of all the instance variables for that class, including instance variables inherited from ancestor classes. The upper-right window contains a list of

The topology of object-oriented programming is laid out in Browser, a window with four child windows.

all methods defined for that class. If a particular method is selected, as in Figure 1, then the source code for that method will be displayed in the larger, lower window.

Getting Started in Actor

As mentioned above, most of the real work in Actor is done in Browser. However, the process of familiarizing yourself with Actor usually begins with some interactive dialogue in Workspace.

The timid programmer may begin by typing the command

```
print("Hello, world")
```

and hitting Return, which will output a line to Actor Display. As you continue in this way, it soon becomes obvious that Actor syntax is fairly straightforward; although Actor has the heart of a threaded interpreter, the cryptic postfix notation normally associated with Forth has been thoughtfully replaced by a more friendly C- or Pascal-like grammar with a very limited number of constructs.

A more impressive demonstration of Actor's capabilities can be obtained by typing the following two lines in Workspace:

Eco-C88 C Compiler with Cmore Debugger

Professionals prefer the Eco-C88 C compiler for ease of use and its powerful debugging features. Our "picky flag" gives you nine levels of lint-like error checking and makes debugging easy:

"I'm very impressed with the compiler, editor, and debugger. I've tried quite a few different compilers for the PC and have given up on all of the others in favor of yours... I've gotten to the point where I download C code from a DEC VAX/VMS system just to be able to compile it with the picky flag set at 9. It finds lots of things VMS totally ignores..."

JS, Oak Ridge, TN

The Eco-C88 compiler includes:

- A full-featured C compiler with 4 memory models (up to 1 meg of code and data) plus most ANSI enhancements.
- Without a doubt, the best error checking you can get. We catch bugs the others miss, making you much more productive.
- Cmore is a full-featured source code debugger, not some stripped-down version.
- Robust standard library with over 230 useful (no "fluff") functions, many of which are System V and ANSI compatible. Full source is available for only \$25.00 at time of order.
- CED, a fast, full screen, multiple-window program editor with on-line function help. You can compile, edit, and link from within CED.
- cc and mini-make utilities included that simplifies the most complex compiles.
- Users manual with over 150 program examples (not fragments) to illustrate how to use the library functions.
- Fast compiles producing fast code.

Our Guarantee: Try the Eco-C88 compiler for \$99.95. Use it for 30 days and if you are not completely satisfied, simply return it for a full refund. We are confident that once you've tried Eco-C88, you'll never use anything else. Call or write today!

Orders: 1-800-952-0472

Info: 1-317-255-6476




Ecosoft Inc.

6413 N. College Avenue
Indianapolis, IN 46220

ECOSOFT

CIRCLE 42 ON READER SERVICE CARD

Browser: Scribble			
DirtyCLD	↑	hWnd	↑
FileDialog		defProc	
FractalDialog		parent	
InputDialog		cRect	
Window		paintStru	
Scribble	↓	hMenu	↓
<pre> /* Initialize mouse dragging. */ Def beginDrag(self, wP, point) { dragDC := getContext(self); if not(saveLines) then savelines := new(OrderedCollection,3000); endif; add(savelines,point); isLBUTTONDOWN := true; } </pre>			

A screenshot of a digital drawing application window titled "Scribble". The window contains a white canvas with the word "Hello" written in a simple, hand-drawn, black-outlined font. The letters are slightly irregular and sketchy. The window has a standard Mac OS X title bar with a red close button, a yellow maximize button, and a green window button.

The screenshot displays the Actor Display environment with the following components:

- Browser: GraphDemo**: The main window containing a code editor and a demos list.
 - Code Editor**:


```

Accept! Edit Doit! Inspect! Options
Utility Templates

Window
  GraphDemo
  TextWindow
  EditWindow
  WorkEdit
hWnd defProc
parent cRect
paintStrud axes
command downScale
draw init
Def draw(self, aBkI | longStr, new_FX);
{ hdc := getContext(self);
  eval(aBkI);
  X := start;
  precision := MaxPositiveInteger / ((stop - start) + 1);
  setMode(self);
  repaint(self);
  axes(self, 0.0, start, 0.0, stop);
  axes(self, start, 0.0, stop, 0.0);
  do (over(asInt(start), asInt(stop)), asInt(precision)) {
    drawLine(hdc, X, 0.0, X, 0.0);
    X := X + 1;
  }
}

Demos List
  Turtle graphics
  Track
  HyperCube
  GraphWindow

```
 - Demos List**: A panel on the bottom left listing available demos: Turtle graphics, Track, HyperCube, and GraphWindow.
- Actor Workspace**: A window on the top right with a menu (File, Edit, Doit!, Inspect!, Browse!, Cleanup!, Show Room!, Utility, Templates, Demos!) and a text area.
- A Graph Window**: A window at the bottom right displaying a sine wave plot on a coordinate system.

The first line creates a new object of the Window class and places its name, "Sam," in the Actor dictionary of global variables. (Object-oriented purists please note: The *defaultNew* method is considered a "class method," so it really acts on WindowClass, a class object created at the time the Window class was defined.) The second line will make this object visible on the screen.

A little exploration rapidly reveals that our new window is the bona fide article: It can be moved, resized, and closed, just like any other window on our Windows screen. Indeed, in Actor parlance, "Example Window" has inherited all the characteristics of the class Window.

Just what has happened to create Example Window can be explored in detail via Workspace, Browser, and Inspector. What's particularly nice about Actor is that we can do our exploring while Example Window remains resident. We can even change the behavior of Example Window by either altering its instance variables or invoking some of its current methods. In fact, we can also change these methods or add new methods, which will instantly affect the behavior of Example Window.

Digging Deeper into Actor

To explore Actor further, we modified and expanded on an example provided with the Actor tutorial called "Scribble." Scribble is a type of window in which the user can use the mouse to scribble to his heart's content (Figure 2), then erase the scribble or redraw it. While Scribble is a very small affair, it does show how little Actor code is needed to create a simple, interactive application. Listing 1 shows a Scribble class file, SCRIBBLE.CLS, after our changes. Although the class file is typically constructed using Browser, and is thus seldom viewed in this form, it is useful to see it because it shows the entire contents of the class in a single glance.

JULY 1988

The object of the inherit method is *Window*, which indicates that *Scribble* will be a descendant of *Window* and inherit the methods and instance variables of the *Window* class. The inherit method also defines the name of the new class, *#Scribble*, and the names of the new instance variables we will be adding to this class.

The new instance variables for this class consist of *dragDC*, which will contain a display context for the mouse drawing activities; *saveLines*, which will be used to store the drawing as a collection of points; and *isLBUTTONDOWN*, a flag that indicates whether the left mouse button has been pressed.

Next comes two *now* methods that provide context switching for the Actor compiler. Following these are the definitions of the new methods for the *Scribble* class. Each new method definition starts with the *Def* label, followed by the method name, and then the parameters used by that method in parentheses. The first parameter is always *self* (the object to be acted upon), followed by the parameters to be passed to the method when it is called. If the parameter list contains a *|* character, the parameters after the *|* are treated as local variables only.

You may have already have noticed that the first two methods, *WM_MBUTTONDOWN* and *WM_RBUTTONDOWN*, look suspiciously like Windows messages. This is no accident. If Actor receives a Windows message, it automatically routes that message to a method of the same name in the class of the appropriate receiving object. If no method of that name exists for that object's class or ancestor classes, the message is ignored. This is the way Actor implements the message processing loop normally required when programming for Windows in other languages.

In this file, we have defined the behavior activated when the middle and right mouse buttons, respectively, are pressed. When the middle button is pressed, the *WM_MBUTTONDOWN* method executes the method *repaint*, which is predefined here in the ancestor class *Window*. In this case, *repaint* clears the window. If the right mouse button is pressed, the window is erased.

After *repaint* is executed, the *WM_MBUTTONDOWN* method goes into a *do* loop, going through all the elements of the collection of points called *saveLines* and drawing lines between the even and odd numbered points. The *moveTo* and *lineTo* methods used here are defined in the

Point class provided with Actor.

The *WM_RBUTTONDOWN* method is similar to *WM_MBUTTONDOWN*, except that instead of redrawing the scribble from *saveLines*, it resets *saveLines* to *nil* (empty).

The functions of the other three methods should now be relatively easy to puzzle out. The *beginDrag* method starts by asking Windows for a display context (for drawing lines) using the predefined method *getContext*. It then initializes *saveLines* if required, adds the current cursor point to *saveLines*, and sets the *isLBUTTONDOWN* flag to say that the left button is placed. By the way, an *OrderedCollection*, like *saveLines*, is an instance of a predefined Actor class that is similar to an ordinary array.

If the left button has indeed been pressed, the *drag* function will *moveTo* the last point in *saveLines*, draw a line between it and the current point, and then add the current point twice to *saveLines*—once to store the end of the last line, and again to store the beginning of the next line.

Finally, the *endDrag* method releases the Windows display context

received in *beginDrag* using the predefined method *releaseContext*, it adds a final point to *saveLines*, and it turns off the flag *isLBUTTONDOWN*.

If you've followed this so far, you probably have at least one question: How are the *beginDrag*, *drag*, and *endDrag* methods ever invoked? The answer is simple but not obvious. These methods are respectively called by the *WM_LBUTTONDOWN*, *WM_MOUSEMOVE*, and *WM_LBUTTONUP* methods predefined in the *Window* class, which is the ancestor of *Scribble*. This is a good example of how inheritance can be used to "hide" the Windows-dependent part of an application.

Evaluating Actor

Obviously, our example gives only a small taste of the power of Actor. During our six weeks with Actor, we wrote working, if somewhat crude versions of a script file interpreter, a spreadsheet, and an electronic mail system. We've seldom found ourselves so productive so quickly in any new programming environment.

Perhaps the biggest single benefit of Actor is that it allows you to learn

WE GET INTO THE BEST PLACES



You're in good company when you slide the ACL™ multiuser board into your PC. Sixty percent of the Fortune 1000s use them to connect terminals, printers, plotters, modems, and other PCs to single personal computers. Simple. Low cost. Compatible with most multiuser systems. Find out why major companies use the ACL. You'll discover that they have a good reason.

CALL: 1-800-STAR GATE

Star Gate Technologies, Inc.
29300 Aurora Road • Bentley Park South
Solon, Ohio 44139 • Telefax No. (216) 349-2056

STAR GATE
TECHNOLOGIES INC.

Shake Hands With Tomorrow

CIRCLE 81 ON READER SERVICE CARD

Windows interactively. Actor allows you to make direct calls to Windows, and the predefined Actor code has numerous examples of how to use many of the most commonly used calls. Windows is a very large and sophisticated environment that takes a long time to master in any environment, but working with Actor is far more convenient and rewarding, because each Windows option can be tested "on the fly." Furthermore, Actor manages to offer this high level of interaction and still execute at very acceptable speeds.

Actor is an excellent introduction to object-oriented programming. While most other object-oriented programs tend to be either resource hogs that work well only on souped-up hardware, or clunky preprocessors that aren't interactive, Actor gives the user a fairly sporty ride through this new programming genre on an ordinary PC or AT.

Actor also contains a lot of interesting tidbits, such as a complete system for building your own command languages that's based around the Yacc (Yet Another Compiler Compiler) utility commonly associated with UNIX. And users who just want to root around the Actor classes that are already provided can learn a lot about creating such applications as editors and parsers.

Caveats

At \$495, Actor is clearly a good way to get a head start on Windows. However, Actor does have some implementation problems that may

annoy both users new to Windows and more experienced Windows programmers.

For new users, the principal problem with Actor is its somewhat limited set of application development tools. Most of the predefined methods were created to support the

While most other object-oriented programs tend to be resource hogs, . . . Actor gives the user a fairly sporty ride . . . on an ordinary PC or AT.

Actor tools themselves, so there are some gaping holes. For example, there are almost no methods having to do with graphics. Users are left on their own to figure out how Windows handles icons, bit maps, and other graphics objects. Similarly, there is no direct support for communications or print spooling.

Generally speaking, the Actor

classes provided were written to support the Actor tools, and little else. This can be very frustrating, but Whitewater justifies this by noting that more space is therefore available in a crowded 640K area for the user's own creations.

For the more advanced Windows user, Actor presents a different set of problems. Actor may still be suitable for modeling applications, or for generating smaller applications quickly, but advanced users are likely to resent their inability to directly control memory and real-time event handling as they can in a lower-level environment such as C. In addition, they have to deal with Actor's high overhead, which certainly won't allow the creation of space-conscious stand-alone Windows utilities such as Clock or Clipboard.

Similarly, the absence of a full-featured, low-level debugger in Actor is a definite disadvantage for more sophisticated users. Although the regular Actor debugger is useful, it responds only to breakpoints embedded in the code and doesn't extend below the Actor code level. Version 1.1 comes with a somewhat lower-level debugger that outputs to a terminal off COM1:, but the manual warns that this feature may not be supported in future versions.

Actor also does not offer anything like the *Make* facility available for most compilers. Since the process of rebuilding an Actor application must take place within Actor itself, no commercial *Make* facility can be used. However, as your Actor code

Listing 1. Class file for the Scribble Class

```
/* Scribble is an example in the Actor tutorial
which we have expanded to show we worked with Actor. */ !!

inherit(Window, #Scribble, #(
  dragDC
  saveLines
  isLBUTTONDOWN
), 2, nil)!!

now(ScribbleClass) !!

now(Scribble) !!

/* Window's middle-button-down message. Redraw scribble
picture pressed. */
Def WM_MBUTTONDOWN(self, wp, lp) temp, dragDC
{repaint(self);
 dragDC := getContext(self);
 do(saveLines,
  {using(point)
   if not(temp)
    then temp := point;
    else moveTo(temp, dragDC);
    lineTo(point, dragDC);
    temp := nil;
   endif;
  });
 releaseContext(self, dragDC);
} !!
```

```
/* MS-Window's right-button-down message. Erase
the window if pressed.*/
Def WM_RBUTTONDOWN(self, wp, lp)
{ saveLines := nil;
  repaint(self);
} !!

/* Initialize mouse dragging. */
Def beginDrag(self, wp, point)
{ dragDC := getContext(self);
  if not(saveLines)
    then saveLines := new(OrderedCollection, 3000);
  endif;
  add(saveLines, point);
  isLBUTTONDOWN := true;
} !!

/* Initialize mouse dragging. */
Def drag(self, wp, point)
{ if isLBUTTONDOWN
  moveTo(last(saveLines), dragDC);
  lineTo(point, dragDC);
  add(saveLines, point);
  add(saveLines, point);
  endif;
} !!

/* Initialize mouse dragging. */
Def endDrag(self, wp, point)
{ releaseContext(self, dragDC);
  add(saveLines, point);
  isLBUTTONDOWN := nil;
} !!
```


proliferates, the absence of a *Make* could become increasingly frustrating, and right now Actor users must write their own.

Another problem that may irk some developers is the fact that there is no way to distribute Actor add-ons except in source code. Whole Actor applications can be sealed off and shipped without source, but there is no encoded form of source file that can be shipped to other Actor users to be incrementally added to their existing Actor environment. This effectively forces third-party supporters of Actor to give away their source, whether they want to or not.

Documentation and Support

Since we cut our Actor teeth on the earlier and murkier Version 1.0 manual, we are very glad to report that the documentation for Version 1.1 is much improved. The manual includes a tutorial, an overview guide to the supplied Actor Classes, a section on advanced topics such as the YaccMachine, and a step-by-step guide to creating stand-alone applications from Actor. The manual also includes an exhaustive compilation of the supplied Actor classes, a list of all Microsoft Windows calls, and a good index.

In general, we found the manual to be a good introduction and reference guide for later on, but it's a little murky in the middle. The addition of a more advanced tutorial covering the creation from scratch of one or more applications, such as a simple spreadsheet or card file system, would make the manual a lot more useful.

Support is available from Whitewater under two plans: Plan 1, at \$100 per year, allows a limited number of direct phone calls to technical personnel and unlimited access to Whitewater's Technical Support BBS. Plan 2, at \$200 per year, allows unlimited access to both. Whitewater also offers classes in Actor at its offices near Chicago.

Whitewater is still a very small company, so even a relatively small number of calls to technical support will eventually put you in touch with most of staff. This can be an advantage, since everything is coming from the horse's mouth, or at least close to it. Our experience with technical support was very positive; they were patient and had good technical insight.

Frankly, we hope Actor and Whitewater will be around for a long time. As Microsoft's Windows tools im-

prove, we suspect that Whitewater's Actor may not remain as compellingly attractive for new users as it is today. However, it is not difficult to imagine a growing base of Actor users among those who recognize the benefits of being able to tinker with Windows in a simple, interactive environment, without having to deal with the very demanding details of interfacing to Windows itself. □

Michael Guttman is Director of Consulting Services with Morrissey Associates, a software development and consulting firm with offices in Skokie, Illinois, and Dallas, Texas.

Did you find this article particularly useful?
Circle number 6 on the reader service card.

Product Information

Actor \$495

The Whitewater Group
Technology Convention Center
906 University Place
Evanston, IL 60201
(312) 491-2370



Serious programmers look for the Programmer's Power Pack—

the key to finding the programming tools and utilities that they need to get ahead in their work.

Programmer's Power Pack is the one card deck no serious programmer can live without...and it's free!

Look for the Programmer's Power Pack in your mailbox...it's coming soon.

"Print it again, Sam"



Spread the good news about your product with a Micro/Systems Reprint

You can have your article or review professionally reprinted on 60 lb. paper stock, in 4-color, 2-color or black & white. Call today for more information.

Shaun Hooper
(415) 366-3600

* Minimum quantity 100 reprints

Novell NetWare and Banyan VINES/286

Advanced NetWare may provide a solution for single-server systems, where VINES/286 is best suited for multiple-server systems.

by William Wong

Novell NetWare is one of the most popular PC LAN operating systems. It has a number of advanced features, including various file protection levels, multiple server support, and support for diskless workstations. NetWare comes in 8086, 80286, and fault-tolerant versions. The 80286-based Advanced NetWare is the most common installation and the version reviewed here was run on a Standard Microsystem Corporation (SMC) twisted pair Arcnet system.

Banyan VINES is also part of a family of network operating systems. VINES runs on the 80286, 80386, and 68000. Banyan sells its own dedicated servers based on the 80386 and 68000. VINES is also sold for generic 80286 and 80386 systems. VINES/286 was also reviewed with SMC Arcnet.

The configuration examined is designed for a medium-sized workgroup with a dedicated 80286 server. Ten to twenty users is typical for this size system, which is appropriate for Arcnet, an 80286 and NetWare, or VINES.

Novell NetWare

Novell Advanced NetWare 286 is one of the most popular network operating systems available for PC-compatible systems. Advanced NetWare is a server-based system, although a server can also run as a DOS workstation. NetWare supports a variety of network adapters, includ-

ing Arcnet cards from SMC. NetWare 286 runs on both 80286 AT-compatibles and 80286 servers available from Novell.

The server software is based on a proprietary operating system that runs in protected mode on the 80286. A different version of NetWare is available for the 8086, which does not run in protected mode. The protected-mode operating system provides a number of advantages over a DOS-based server, including access to more memory and protection between multiple tasks within the server. Server-based systems are also easier to maintain than server/workstation systems because they can be isolated and do not have to be restarted when a workstation application goes awry.

Advanced NetWare runs on most 80286 AT-compatible computers but not PCs using the small turbo 80826 cards that do not provide support for extended memory. NetWare supports multiple network adapter cards in the same computer, and the adapters can be of the same or of different types. It is possible to mix Arcnet stations with Ethernet stations by hooking up the appropriate station to the corresponding adapter card in the server.

Advanced NetWare also can support multiple servers on the same network. Servers can be connected through any network adapter that

may include workstations on the same link. A server can support shared hard disks, printers, and dial-in communication links.

NetWare support on a workstation consists of a redirector module with support for a network adapter card. The redirector is a TSR program that must be loaded before the network can be accessed. The redirector installs itself between DOS and any applications. All DOS accesses are channeled to either the local copy of DOS or across the network, depending upon how the user configures the system. This allows a mix of local and remote peripherals.

The documentation that comes with NetWare is impressive and usable. It is divided into a number of different manuals, including books on installation, maintenance, server applications, and workstation applications for both the network manager and network users. A cross-reference is included to help a new network manager find out where information can be found. Even so, the number of manuals can be overwhelming for a first-time installer.

Installation

Installation takes time and is nothing like installing Windows or Lotus 1-2-3. Two days should be reserved for an initial installation, although an expert can probably get things

up and running in a few hours if all goes well and a few short cuts are taken.

The documentation is straightforward for first-time installers. It starts with the creation of the NetWare operating system software for the server and for the workstation—two different programs. It is relatively easy to create for a single hard-disk server with one network adapter card, but it gets a bit more confusing when multiple pieces of hardware are installed. The longer installation time on the server is needed because the NetWare installation software wants to perform a low-level format on the hard disk and then run a comprehensive surface analysis. This can run unattended and overnight, which is the reason for the two-day installation time. The process *can* be skipped, but it is well worth the effort and can avoid maintenance and support headaches that can occur later because of disk faults. In general, the hard-disk setup can be done while you are reading the rest of the installation documentation. Adding other adapters and disks is a bit more complicated and requires regeneration of the server operating system.

NetWare requires a keycard or a Novell disk controller card that contains the key. This form of copy protection is used for all Novell servers. The stand-alone keycard takes up one slot, which is usually negligible for an AT with eight slots, but may limit expansion if a PC or small AT cabinet having four or five slots is used. The keycard is matched to the installation disks.

Checking out the network adapters and connections to workstations can be done before or after setting up the server, but preferably before. Diagnostic programs check for the integrity of the adapter card as well as its ability to send information between workstations. Unfortunately, the NetWare diagnostics are designed for use on workstations, and the Arcnet card on the server may be set up differently. This requires testing the server card in a workstation first and then moving it to the server.

Setup of initial shared directories, system attributes, and users comes next and is automated for the network manager, who then must set up the rest of the system. A good understanding of NetWare's mechanisms is necessary to properly set up the system. NetWare divides users into groups and into a number of lists, which are also provided. Protection mechanisms are based

upon users, groups, and lists. Disk protection is available for directories and files. The protection mechanism is quite flexible and program files can be marked for execution only. Users can be given some control over their own directories and files by the network manager.

Disks can be partitioned at installation time and subdirectories provide organized access to the disk. NetWare uses the MAP command to set a logical drive to a partition and a subdirectory in a fashion that is similar to the DOS SUBST command. Multiple logical drives can map to

NetWare uses server-specific references in some of its naming conventions.

the same subdirectory. A minor confusion can arise because you can change the default directory of a logical drive to the root directory, in which case you are in the root directory for the network disk, where you can see all the subdirectories and files in the root. This is the default installation setup and can be changed by the network manager using the protection mechanism.

Usually, a network manager will set up a user's login configuration to specify the logical devices that are available. NetWare, however, uses server-specific references in some of its naming conventions. This can make things more difficult for network managers and advanced users in a multiple server network. Some problems can arise if servers are added or services are moved from one server to another. For example, a disk service may be moved from one server to another with unused disk capacity. User configurations must then be changed to reflect the change in servers.

Features

Most services are available or controllable through programs that are menu-driven with context-sensitive help, or through command-line op-

tions. The latter is useful when creating batch files to perform special functions, such as automatically sending mail or changing the setup of devices. The menu interface uses popup windows but it is not always apparent what keys and options are available without referring to a manual. For example, adding an element to a list of items is done using the insert key. Help is accessed using the F1 key, and a limited natural language help facility is provided by entering questions. The main problem with the latter is that you need to know or discover the right questions and syntax.

Novell's Advanced NetWare does come with a number of useful network utilities that are sometimes options from other network software vendors. Electronic mail is standard. It is command-driven with context-sensitive help. The mail system is flexible but more difficult for casual users than a menu-driven system. Data files also can be sent as attachments to messages. A remote dial-in capability is supported using standard serial ports. This requires an additional workstation software configuration. The same is true for automatic workstation loading using boot ROM on a workstation's network adapter card. The procedures outlined for this installation are well documented for each type of network adapter. The basic setup describes how to get DOS loaded, but it is possible to load any number of device drivers for EMS RAM, RAM disks, and special hardware that may be installed on a station that boots from ROM. Third-party vendors provide additional services, such as dial-out and 3270 terminal links. These options are not part of NetWare.

NetWare is somewhat unusual because it comes with a multiuser network game called Snipes. This is a maze game in which all users play from their own PCs attached to the network. You can encounter other users in the maze. It is not the greatest game in the world, but it shows what can be done on a network, and it can be addictive.

Performance

Advanced NetWare Version 2.0a was tested on a 10-MHz, 80286, AT-compatible server with a 40-MB hard disk connected to a number of different workstations using SMC twisted pair and coax Arcnet products. This configuration will easily support a dozen workstations and NetWare/Arcnet systems of 30 or 40 workstations, depending upon

network utilization. A larger number of word processing or spreadsheet applications can be supported than can database or program development workstations. Response time tends to rise linearly at a low rate for each additional active workstation. NetWare handles small files and record requests at the same transfer rate as larger files when programs or workfiles are loaded. Network database applications using either NetWare's proprietary semaphores or DOS 3.x's file locking run very well on this system.

Multiple hard disks with capacities in the hundreds of megabytes can be supported on a single server. A server can run with as little as 10 megabytes of disk storage, but the practical minimum is 30 megabytes. Backup can be done using floppy disks, but third-party tape systems are available for backup that is normally done through workstations attached to the server. Most tape backup software supports NetWare's protection, and a password system allows a network manager to back up or restore files with their original protection intact. Only server file backup can be done over the network, not workstation files.

Over all, Advanced NetWare 286

is a solid, high-performance, server-based, network operating system that works in single- or multiple-server configurations. It runs on 80286 and 80386 AT-compatible computers and provides a number of services in a transparent fashion. Theoretically, one server could support up to 1,000 workstations using Arcnet adapters (four adapter boards in the server, with 250 stations attached to each through hubs). The network management utilities are flexible but somewhat cryptic in their operation until you get used to them. User utilities have similar problems that can cause difficulties for new or casual users. Installations that rely on electronic mail should investigate one of the many third-party electronic mail packages for NetWare. Server backup is not integrated as part of the system but the variety of good third-party solutions makes this problem inconsequential. Multiple server environments are very practical, but services are server-specific, which means users must know the server topology and when it changes.

Advanced NetWare will work with one megabyte of memory but two megabytes is more desirable. The additional memory is used for buffer space that improves system perform-

ance. Adding memory is one way of upgrading a server. However, the operating system must be set up to use newly added memory.

Advanced NetWare and Arcnet make an excellent fit for small- to medium-sized workgroups that require one or two servers. Arcnet provides a low-cost solution allowing diskless workstations to be used. The cost for a low-end, monochrome, 8088 diskless NetWare/Arcnet workstation is comparable to a multiuser system terminal.

Novell also has a number of other network products, including NetWare SFT, a fault-tolerant version that supports disk mirroring and transaction oriented operations. An 80386-specific version will be available later this year. It should provide an even higher performance server platform than the 80286 version. It should also provide more mini- and mainframe-level access and audit controls, such as limiting the logon time to the network to 8 am through 5 pm, or forcing you to change your password on a weekly basis.

Banyan VINES/286

Banyan VINES is an alternative to Novell NetWare. VINES stands for Virtual Networking System, and the

9-Track Tape Subsystem

The Data Interchange Solution You've Been Waiting For!

Qualstar's 1/2" 9-track Ministreamers™ bring full ANSI data interchange capability to your PC or Macintosh™ computer system.



With 9-track tape, you are free to exchange data files with any mainframe or minicomputer in the world.

Our affordable Ministreamers come in both 7" and 10 1/2" versions and use less desk space than an ordinary sheet of paper. They can provide 1600 thru 6250 BPI capability and may be used for disk backup as well as data interchange. Complete subsystem prices, including software, start as low as \$2,495 for 7" units and at \$3,670 for 10 1/2" units.

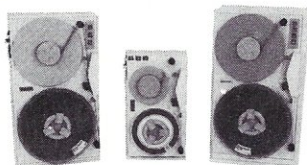
Qualstar has become the market leader in desk-top 9-track systems for a good reason; our tape drives have established an outstanding record for reliability and low cost of ownership.

Discover the many advantages 9-track tape has over other Micro/Mainframe links.

Call us today!

QUALSTAR®

9621 Irondale Avenue,
Chatsworth, CA 91311
Telephone: (818) 882-5822



Macintosh is a trademark of Apple Computer, Inc.

CIRCLE 67 ON READER SERVICE CARD

SELF-INKING PRINTER RIBBON

**Awarded United States
Patent #4701062
Lasts 10-15 times longer
than the conventional ribbon.
For printers using 1/2" width
open spool ribbon:**

- Okidata-82A-83A-84-92-93
- Teletype-33, 35
- Star Gemini 10X
- Extel
- Dec LA 180/120
- Dec LA 30/IBM 1443
- Teletype-Model 40
- Texas Instrument 800/810, 820, 880

CONTROLLED PRINTOUT DEVICES, INC.

P.O. Box 869, Baldwin Road
Arden, NC 28704

(704) 684-9044 • Telex: (Filmon-Aren) 577454
Contact us by mail, phone or telex and we will forward you a brochure

CIRCLE 64 ON READER SERVICE CARD

banyan tree, which is Banyan's logo, has vines. VINES runs on an 80286 AT-compatible, and is available on 68000-based servers from Banyan. Banyan has also started delivery of its new 80386-based server, 80386 upgrades to its 68000-based servers, and a version of 80386 computers. VINES/286 v2.1 is reviewed here. It is the low-end version intended for medium to large networks. VINES excels in multiserver LANs and WANs.

The capabilities of VINES are similar to Novell's NetWare. Each system has its own advantages and disadvantages with respect to the other. Although not reviewed here, the servers available from Banyan have one major advantage over VINES/286 and Novell NetWare—they come with a built-in DC-600 tape drive, used for backup as well as software installation, that makes a major difference. In fact, the price difference between these servers and the 80286-based products can often be justified by the cost of installation and maintenance alone.

VINES uses a three-level, global naming facility called StreetTalk to provide transparent access to services, regardless of where the service and user are located within the network. The three level names are given as *Item@Group@Organization*. Items are users, lists, disks, printers, and any other service or object supported by the network. The StreetTalk naming system is distributed among servers in a multiple server network. Services for a particular group also can be spread across servers. In fact, services can be moved from one server to another without a user knowing that they have been moved. However, a service must be turned off before it can be moved, which normally means moving things at night or on weekends when the service will not be used. Nicknames or aliases are supported by StreetTalk so different users can access the same service using different logical names. StreetTalk names are also the basis for protection mechanisms. Wild card options similar to DOS are used so providing unrestricted access to all users would be specified by **@*.**. The flexibility and consistency of StreetTalk is one of the most powerful features of VINES.

Although hidden from view, VINES does run on UNIX. This is true for VINES/286 as well as the Banyan servers based upon the 68000 and the 80386. UNIX provides a flexible and consistent platform for VINES, but it takes up a good deal of memory and disk space compared to other

network server software like NetWare or Tapestry. However, the added requirements are insignificant and are easily outweighed by the power and flexibility of VINES. The difference is about 0.5 to 1 megabyte of RAM and a few megabytes of disk space make little difference when the hard disk has more than 40 megabytes.

VINES documentation is divided into three major books: installation guide, manager's guide, and user's guide. The books are augmented by supplements describing new features. A good index and separator tabs

*Unlike disks
in NetWare,
VINES logical
disks are
distinct and can
be created
and deleted as
necessary.*

make finding information quick and easy. The VINES user interface is divided into three similar programs: system console program, network manager's program (MANAGE) and user's interface program (VINES). Each is menu driven. The latter two provide context sensitive help via the F1 key.

Installation

Installation takes three to four hours and is similar to a Windows-like setup program. It asks questions about the hard disk, the installed network adapter, and what printers are attached. A surface analysis of the hard disk is also done that takes about half an hour. The entire system is installed directly on the hard disk. The VINES server starts up after you tell it its name and the network manager's StreetTalk name, group, organization, and password. Options such as electronic mail then can be installed.

VINES can use partitioned disks, but VINES also allows a disk partition to be divided into multiple logical disks. The disks work like subdirectories, where free space from the disk partition is allocated to each logical disk as necessary. However, unlike disks in NetWare, VINES

logical disks are distinct. They can be created and deleted as necessary.

VINES/286 uses a server key that attaches to the printer port. A printer can be attached to the other end of the key so the printer port is still usable. The key contains the list of enabled options and is read when VINES is loaded. Option keys are attached to the server key during the option installation process. The keys contain non-volatile memory and enabling an option causes the serial number to be copied from the option key to the server key, after which it is deleted from the option key. The option can be uninstalled in the same way, and the option key can be used again on another server or reinstalled. VINES software includes all options but only enables those found in the server key. This makes purchase and installation of options very simple, but does use hard disk space for options that are not enabled. However, the amount of space occupied by option programs is only a few megabytes at most. The option support is menu driven and the installed option in server and option keys can be viewed using this support function.

Network adapter and printer support is controlled in a similar fashion. Cards and ports are added and deleted after adding or before removing a card or printer. Appropriate drivers can always be loaded, so it is simply a matter of selecting the corresponding item from a menu to perform these operations. Memory utilization is automatic and VINES uses all extended memory available. Two megabytes is the minimum configuration, and three or four is more practical for larger installations.

Setting up a new user is relatively easy. The user's capabilities, password, and initial login services are set up from menus in the MANAGE program. MANAGE also controls peripherals, services, and StreetTalk lists and nicknames. A user can be given a nickname when a user's account is set up. The PCCOPY and PCCONFIG programs are used to set up a workstation disk. Software options include up to a three-group search list. This allows a user to log on using a short nickname like WW instead of *WilliamWong@Software@LANSystem*. Support for options such as NETBIOS emulation are done using PCCONFIG. These options can be changed at any time. The primary difference between NetWare and VINES here is that NetWare only needs two files (three for NETBIOS support), while VINES uses half a

dozen. However, PCONFIG is a menu-driven program that allows a user to modify the configuration, while NetWare requires a number of configuration disks and normally requires a network manager to do the changes.

Features

User-support utilities can be accessed from the VINES menu program. Disk selection, printer support, and password maintenance are supported, along with mail and communications support, if enabled. The MANAGE program provides menu access to network management programs to control network services. Both programs start up other VINES programs to perform functions, and the name of the program is displayed while it is loading. The programs can be invoked directly so a user can either start VINES and select electronic mail or simply run the MAIL program. The functions of the network management tools are controlled by the *Adminlist* list associated with the corresponding service. A user may see if a service is running but may not control it unless his name is in the *Adminlist*. Multiple lists can exist and each server and services will have its own lists. This

permits you to manage your own servers, or it allows a single network manager to control all servers and services from any workstation.

Unlike NetWare, the VINES electronic mail package is provided as an option and is somewhat expensive. A single mail package can support multiple servers, but multiple mail servers are possible and desirable with a large number of users. The program is menu-driven with context-sensitive help. The recipients of a message can be typed in or selected from a list of StreetTalk names. Files also can be attached to messages. VINES will hold mail if the server supporting the recipient user is not connected to the network. Mail is exchanged between servers when two networks are connected together. This situation occurs quite often when VINES is configured into a WAN, which is occasionally done via modems. It also has a failsafe capability that allows mail to be sent to someone even though the link between servers may be temporarily disconnected.

NETMAN is another VINES option. It provides access to network statistics and can print reports about users, services, and the operation of the network links. NETMAN enables

a network manager to check the loading on each service to see how resources should be allocated or increased. It is well worth the cost and only a single copy is needed to support a network of any size.

The communication options are quite powerful. They do require the addition of an intelligent communications controller from Persyst. The board supports up to four communication lines that can be directly connected to workstations, servers, modems, or host computers. The lines can be used for dial-in support, server-to-server connections, asynchronous terminal emulation, or synchronous connections to a host computer using SNA 3270 emulation. The latter can support multiple sessions on a single line. The system console is used to configure each line and each line must be dedicated to a particular service. However, lines having the same service can be pooled so multiple modems can be shared by everyone on the network. Intelligent modems are supported from low-speed modems such as Hayes through high-speed modems such as Telebit. Server-to-server links can utilize the high-speed modems as well as X.25 links. VINES can be set up to call another server at a particu-

€ PROGRAMMERS! THE TOOLS YOU NEED AT A PRICE YOU'LL LIKE

BTree

Supports all index file operations. Very quick sequential or random access, duplicate keys, multiple indices, fixed and variable length data records are all supported.

75.00

ISAM

Works on top of BTree to provide a simple, yet powerful application program/file system interface. Complex filesystem manipulation becomes a snap. Provides the power of a database manager with the flexibility of a programming language.

40.00

lp

Finally, a completely device independent printer library! lp drives any printer as accurately as possible and allows easy access to its most sophisticated features. Multiple fonts, multi-column output, complex margin formatting, and much more. Pays for itself the first time it's used.

75.00

Snake

The ultimate 'make' utility. We couldn't find a good one, so we wrote a great one. Has all kinds of powerful features including wild card filename expansion, nested macros, and multiple dependency and rules definitions. Ready to go for MS-DOS; C source is there if you use another operating system.

59.00

Combine & Save: BTree + ISAM + lp **159.00** + Snake **199.00**

Each product includes a typeset manual, example programs, and complete C source code that runs on any operating system. Softfocus products may be incorporated into applications royalty-free.

Credit card orders accepted. Visa, M/C, Amex. Dealer inquiries invited.

CALL ADVERTISER DIRECTLY

**NOW
MULTI-
USER
AVAILABLE
60.00**

softfocus

1343 Stanbury Drive
Oakville, Ontario, Canada
L6L 2J5
(416) 825-0903

lar day and time and break the connection after a specified period of time. Electronic mail is automatically exchanged after a connection is made, and VINES remembers the connection, allowing mail to be set up for subsequent exchange at a later date.

The asynchronous terminal support works like a conventional communications program, but it only supports Kermit file transfers, not the more popular XMODEM protocol. VINES Version 3.0 will have Crosstalk-compatible script file support as well as XMODEM file transfers. Received text can be saved in a file or sent to the printer. Text files also can be sent without a file transfer protocol.

Performance

VINES server performance is adequate for small to medium workgroups. VINES disk access is comparable to Advanced NetWare for large record sizes (4K and larger) and files. However, VINES takes more time than NetWare to transfer smaller record sizes (under 1K). Most applications are not dependent on small records but multiuser database applications usually do because of record locking considerations. The larger Banyan servers may be required for adequate performance in this in-

stance. Increasing the amount of RAM helps a great deal and keeping the number of services to a minimum helps, too. VINES does not support diskless workstations at this time but may do so in the future.

A VINES network can consist of more than one server of various types.

Backup must be done using floppy disks. All services can be saved or restored to disk but it may be more appropriate to back up files using a tape unit attached to a workstation. However, protection information will not be saved using this approach because VINES does not enjoy the

third-party support that Novell has in the tape backup community. Banyan servers are much more sophisticated because they include DC-600 tape drives instead of a floppy disk drive. The tape unit can be used for automatic or manual backup that is incremental or complete by server and service.

A VINES network can consist of more than one server of various types, so VINES/286 systems can be combined with Banyan servers. The next release of VINES, Version 3.0, will add new features including more comprehensive protection, audit controls, and workstation printer support so shared network printers can be attached to servers or workstations.

Summary

Novell Advanced NetWare 286 is a great partner for Arcnet. Arcnet provides the performance necessary for small to medium workgroups. NetWare is optimized for single 80286-based server systems for which it does an excellent job. As a disk server, NetWare 286 is a better selection than VINES/286 from a performance standpoint. Although NetWare does support multiple servers, it is less transparent and less flexible than VINES. For example,

MICROSOFT, TURBO AND MIX POWER C PROGRAMMERS... C WINDOWS TOOLKIT PUTS YOU IN CHARGE OF VIDEO!

C Windows Toolkit is the only C programmer's windowing package that comes with a complete tutorial on monochrome, Hercules, CGA and EGA video. We don't just provide the functions, we also explain how to use them reliably.

And C Windows Toolkit comes with full, commented source code (would you trust a package that didn't?).

WINDOWING FUNCTIONS

- Create pop-up windows
- Create pull-down menus
- Create spreadsheet menus
- Create context-sensitive help screens
- Store windows for recall later
- Free memory used by windows
- Use 8 different types of exploding windows

SYSTEM SUPPORT

- Detect how many video adapters are present
- Detect the types of video adaptor installed
- Switch between adaptors
- Detect ANSI.SYS
- Control the size and position of the cursor
- Detect the Enhanced Keyboard
- Disable the video signal
- Delay program execution to microsecond resolution

EGA/VGA SUPPORT

- Use all 64 EGA colors
- Use EGA 43-line mode
- Use 2 fonts simultaneously
- Design custom fonts
- Smooth scroll the screen
- Smooth pan the screen

FAST SCREEN I/O

- Write to the screen lightning fast
- Write formatted output (like `printf()`)
- Get snow-free output on the CGA
- Scroll the screen
- Read characters off the screen

HERCULES SUPPORT

- Detect the presence of a Hercules Card
- Detect Ramfont support
- Load a Ramfont
- Switch between modes

Over 80 functions that enhance your productivity.
Full source code included — No run-time royalties
Includes 200 page manual — 30-Day Money-Back Guarantee

Magna Carta SOFTWARE

Requires: IBM PC, XT, AT, PS/2 or compatible that will run Microsoft C and/or Quick C
Supports Microsoft C 4.0/5.0/Quick C, Borland Turbo C 1.0/1.5, Mix Power C

From: **Magna Carta Software**
P.O. Box 475594
Garland, TX 75047-5594
(214) 226-6909

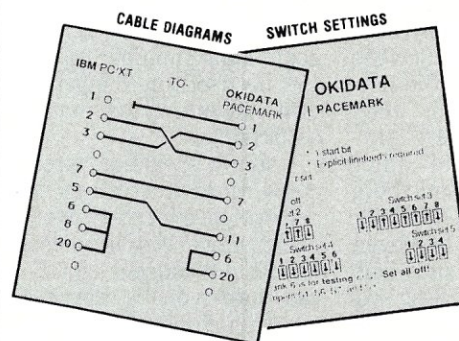


Only \$99.95

(Texas residents add 8% sales tax)

CIRCLE 60 ON READER SERVICE CARD

INTERFACING MICROS



- RS-232-C
- CENTRONICS
- IEEE

Distributors Welcomed!

Made Easy with MICRO-MATCH

A unique product and service for interconnecting

- Micros to Printers
- Micros to Plotters
- Micros to Modems
- Micros to Terminals ...

To hook-up any two of these devices, MICRO-MATCH provides:

- Pin-to-Pin Cable Designs
- Switch Settings
- Jumper Configurations
- Interface Instructions.

Prices begin at \$59 per volume. Send your business card or call for ...

FREE SAMPLES!

COMMAND
COMPUTER CORPORATION

345 Boulevard
Hasbrouck Heights, N.J. 07604

201-288-7000

CIRCLE 6 ON READER SERVICE CARD

NETBIOS support under NetWare uses the NETBIOS driver to allow all PC workstations in one logical network to communicate. However, PCs cannot communicate via a network server to another logical network. VINES, on the other hand, allows any workstation to use NETBIOS or VINES-specific workstation-to-workstation communication support to communicate across the network, regardless of topology. NetWare 286 is less expensive than VINES as a package because many options like mail are included. However, the VINES options tend to be more powerful and easier to use than their NetWare counterparts.

VINES/286 is also a good match for Arcnet in small and medium workgroup situations. However, VINES is optimized for multiple server configurations. The StreetTalk naming system allows a simple network configuration to grow as large as necessary. It also makes upgrades easier, especially when a server is added to increase system performance or when new groups are set up. VINES also supports dynamic control of services while the system is running without reconfiguration. VINES also has the edge for ease of use. All a user really needs to know

is to run the VINES program for network support. Network management is also easier and more sophisticated than NetWare, which does not have an equivalent program for NETMAN.

So which network operating system is for you? Consider Advanced NetWare for single server systems, especially if initial cost or performance are issues. Consider VINES for systems that may grow into multiple server systems. VINES also makes a good candidate with respect to network management and ease of use. If the data is important, then take a look at the Banyan or Novell servers with built-in tape drives. Both NetWare and VINES have the advantage of supporting multiple network adapters so people who need high performance can use Ethernet or Token-ring adapters, while most people can utilize Arcnet adapters.

Although not tested, both NetWare and VINES support a wide variety of network adapters, including Ethernet, Token-ring, Proteon high-speed token-rings, and StarLAN, just to mention a few. Higher performance systems can be configured using the higher throughput network adapters like Ethernet and Token ring. □

William Wong is president of Logic Fusion, Inc., a systems software development firm located in Morrisville, Pennsylvania.

Did you find this article particularly useful?
Circle number 7 on the reader service card.

Product Information

Novell NetWare 286 2.0	\$1,995
4 user starter system	595

Novell Incorporated
122 East 1700 South
Provo, Utah 84601
(801) 379-5900

Banyan VINES/286	\$1,895
Electronic Mail	995
PC Dial-In	1,495
NETMAN	575

Asynch Terminal Emulation	1,295
---------------------------	-------

IBM 3270/SNA (16 Sessions)	2,990
----------------------------	-------

Server-to-Server LAN	1,000
----------------------	-------

Server-to-Server WAN	1,595
----------------------	-------

Persyst Board	850
---------------	-----

Banyan Systems Inc.
115 Flanders Road
Westboro, MA 01581
(617) 898-1000

Banyan Communications Offers True Resource Sharing

Sharing disks and printers on a LAN is the norm but communication services and gateways are the new hot area. Products such as Novell NetWare support gateways to other NetWare LANs, and there is third-party support for shared modems. However, Banyan VINES is one of the few network operating systems that supports all of these facilities in an integrated fashion and does it right, too.

VINES servers need to be equipped with a special communications adapter to provide any communication-related services. The adapter contains an 8088 with its own memory and four serial channels that can be used for any communication service from simple asynchronous terminal emulation to X.25

and 3270 SNA connections. A two-port version is supported and multiple cards can be used on Banyan's own servers for as many as 24 lines.

Using a special adapter costs more but brings a performance increase over using simple serial ports. It does not cause a significant burden on the server's main functions, which typically include disk and printer services. The intelligent communications adapter approach allows multiple high-speed, 19,200-baud modems to be attached to a system, as well as direct connections to terminals or host computers.

Serial connections between VINES servers are augmented by the communication adapter because the adapter handles the

exchange of data between the servers. The server's main processor only needs to exchange data with the communication's processor via shared memory. There is a delay, as data is exchanged between the server's communication processors, but no additional overhead.

VINES provides these services using the StreetTalk naming system, so communication services can be placed on any server in a LAN and be immediately accessible by authorized users. A communication service provides access to modems and other services in a pooled fashion so two or three modems can serve dozens of people who occasionally use a modem.

VINES Version 3, which should be shipping when

this review appears, will have a number of added features on the communications side including the ability to set up the server to link into external mail services like MCI and CompuServe. An MCI gateway is already found on 3Com's 3PLUS system.

Keep communication services in mind when looking into LANs. They can save time and reduce costs when properly implemented, and can often provide services that may be too costly otherwise. For example, a pool of 3270 SNA links is one way to provide mainframe access to all the LAN users without having to include an IRMA card and laying coax for yet another link (you still have to link the PC's up for the network somehow).

Here At Last!

The Authoritative Source of Database Application and Product Information

Inside DBMS Magazine

DBMS Magazine offers in-depth articles examining the capabilities of programmable databases as tools for solving real business problems. Authoritative features show how a specific product is used to solve a specific problem. Sample some of the feature articles to be presented:

- 50 Nasty dBase Bugs, and How to Squash Them
- Has Ashton-Tate Finally Designed a Decent, Multiuser Database?
- Why 386 Databases Can be Either the Fastest, or a Waste of Money
- Is FoxBase+ dFastest Around?
- Why Microsoft and Ashton-Tate teamed up on SQL: Server – What this Ambitious Product Will and Will Not do for You
- R:Base Puts OS/2 to Work

- Understanding Paradox's PAL
- Can Oracle Squeeze Into a PC?

...with regular columns written by experts, providing tips on developing and modifying databases, such as:

- Putting dBase IV's Arrays to Work
- Cross tabulations in R:Base
- Sharing Files Between 1-2-3 and dBase
- Using Clipper's Valid Statement
- Tips on Creating Reports in Paradox

Who Should Read DBMS Magazine?

Professional developers, MIS/DP departments, consultants, VARs and serious users will find DBMS Magazine to be the source of important information for serious users of microcomputer database software.

Charter Offer and Benefits with No-Risk Guarantee

For a limited time DBMS Magazine will offer Charter Subscriptions with substantial cash savings.

You will also be entitled to continued savings — on all renewals and on any gift subscriptions you may order.

If at any time you decide DBMS is not for you, simply cancel. We will send a full refund for all unmailed issues.

Order Today!

Send your order form to:

DBMS Magazine
P.O. Box 3713
Escondido, CA
92025

CHARTER SUBSCRIPTION ORDER FORM

☐ **Yes**, please send my first issue and enter my Charter Subscription to DBMS Magazine. My full year's subscription (11 additional issues) is only \$19.97. I save 44% off the newsstand rate.

SEND NO MONEY!

We will bill you later, after you receive your first issue. GUARANTEE! You may cancel at any time for any reason and receive a full refund on all unmailed issues.

MAIL TO: DBMS Magazine, P.O. Box 3713, Escondido, CA 92025

NAME _____

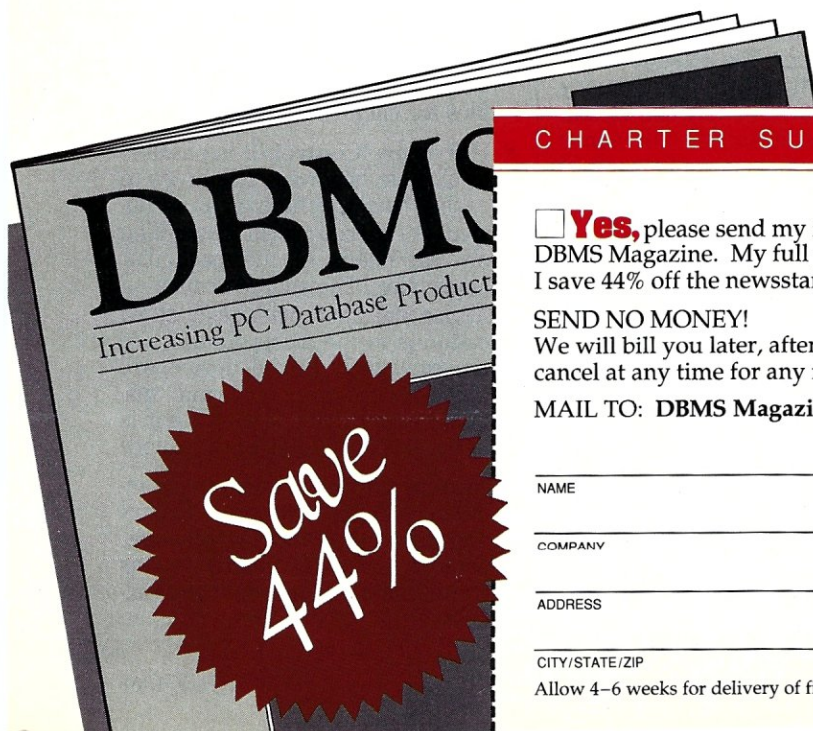
COMPANY _____

ADDRESS _____

CITY/STATE/ZIP _____

Allow 4-6 weeks for delivery of first issue.

Newsstand rate (1 year): \$35.40



VM/386

*An in-depth look at a 386-based
control program to
manipulate multiple tasks for DOS.*

by Charles H. Strom

VM/386 was recently released as a multitasking environment for the IBM PC and compatibles. But VM/386 has a long history. The original development of the product was undertaken by Softguard, the copy protection firm. I suspect that Softguard found that its bread and butter, copy protection, was becoming less popular with software publishers and therefore less profitable. Coupled with this, creating VM/386 proved to be much more formidable a project than originally envisioned. Therefore, Softguard sold the incomplete product to Intelligent Graphics Corporation (IGC). It took IGC six months to complete the project, and Version 1.00 of VM/386 was released at the end of 1987.

IGC bills VM/386 as "the Professional Multitasker." Let's examine the package and my experiences with it over a period of several weeks to see if the company's claims are justified.

VM/386 is a control program for 80386-based microcomputers running DOS. Simply put, VM/386 is a supervisor that directs the creation of multiple, independent tasks, each running a copy of DOS 3.x. Each virtual machine, as IGC calls them, can run unhindered by the actions of the others. A Ctrl-Alt-Del will not affect the entire machine; rather, just that virtual machine will reboot. But I'm getting a bit ahead of myself. Before going into the nuts and bolts of VM/386, we should look at the installation procedure.

Installation

I had an awful time installing VM/386. The manual directs the user to remain logged into his hard disk and type `A:SETUP`, while the installation disk itself directs him to log in the floppy disk and execute the setup

program from there. Several times the installation process locked up the machine, forcing a reboot.

After about two hours of hair pulling, I finally saw the promised installation questions about my hardware (basically categorizing machines into what it called "compatible" and "non-compatible" classes). After getting through this point, VM/386 finally appeared to be properly installed. However, none of the details of the installation process were documented in the manual.

In fairness, I have since received Version 1.01 of VM/386 and had no trouble installing it over Version 1.00. It is unclear if the installation procedure has been fixed or if my problems were a fluke.

Getting Started

Once installed, VM/386 start-up is simple, although care must be taken to remove potentially damaging drivers from `CONFIG.SYS` and `AUTOEXEC.BAT`. Extended and expanded memory drivers, for example, will conflict with the VM/386 memory allocation routines. There is a built-in disk cache, so it makes little sense to include an external cache such as `FLASH`, and I removed this as well. Loading the supervisor is a simple matter: just type `VM/386`, and away you go. After prolonged disk activity, the title screen is displayed, followed by the main VMM (Virtual Machine Manager) menu (see Figure 1).

The manual runs through a clear and thorough tutorial of VM/386 operation. The supervisor is always available and can be accessed by the `SYSRQ` or `Alt-SYSRQ` keys. This brings up a menu that lists active virtual machines (concurrently-running DOS sessions) as well as the VM manager. The manager is structured as a series of nested menus and is very

intuitive. I am the type of user who is probably feared by Intelligent Graphics—I tend to fiddle around with a program before I read the manual, and refer to the manual only when I get stuck. I can't recommend this, but to the credit of the authors, this caused little grief. There is a logic to the structure of the menus that is exemplary, and on-line help is available as well.

Here is a brief summary of the controls available under each of the VMM's main menu:

- **Virtual Machine Control:** This allows creation, termination, parameter updating, and rebooting of virtual machines. In addition, you can customize profiles (memory allocation and a host of parameters related to priorities and hardware allocation) and create `STARTUP` files to permit automatic setup of a configuration upon booting VM/386. Note that this is not the same as `AUTOEXEC.BAT`. Indeed, it is possible to create `CONFIG.SYS` and `AUTOEXEC.BAT` files for each virtual machine.
- **Hardware Control:** It is possible to allocate hardware resources to each virtual machine on an exclusive or shared basis depending on the needs of the particular application. The shared or floating device will link to whichever application is in the foreground, while an exclusively linked device will allow access from a virtual machine, regardless of whether it is in the foreground or background.
- **System Options:** This choice includes screen attributes (colors and intensities) for VM Manager screens and performance options (timeslicing options, debugger access, etc.)

- **Status Display:** This provides a summary of the status of timeslice values, memory information, etc. There is even a memory dump feature that will show RAM contents located in any virtual machine.
- **Help:** There is a very extensive on-line help facility. It is not context-sensitive. However, I don't see any advantage to this type of help facility over the printed manual, which covers the same material in more detail and is easier to use.
- **Exit:** In order to gracefully exit VM/386, all virtual machines are closed and the computer cold-boots.

Benchmarks

Now that I have detailed the operating features of VM/386, the big question is: How well does it perform?

Naturally, any operating environment will exact a cost in terms of the overhead required to manage the system resources. VM/386 is no exception. Norton's SI Version 4.0 reported a computing index of 16.9 versus 17.6 for DOS 3.2, while the disk performance index was 11.4 under VM/386 and 12.8 under DOS. This is not a serious penalty, but things became considerably worse when real-life applications were run.

Computing power was not dramatically impaired, but disk operations suffered severely. Using Wordstar to scroll through a document under DOS (using the Ctrl-QC command) took 1.5 seconds, and under VM/386 took 2.5 seconds, a 67 percent penalty. Loading Ventura Publisher Version 1.1 from disk took

5.5 seconds under DOS and 32 seconds under VM/386. This represents a difference of more than a factor of five! Certainly, this is an area of great concern, and severely cripples this product.

Compatibility

In contrast to the poor showing VM/386 makes in the disk-access-speed department, compatibility is a much happier story. I could find few

*The manager
is structured
as a series
of nested
menus and
is very
intuitive.*

problems using a broad variety of spreadsheet, word processing, desktop publishing, and utility software. My communications program of choice, Pro-Yam, hung—presumably because it interrogates all serial ports on boot-up and I did not link them to the virtual machine in question. Procomm and Qmodem worked fine, however. Some EGA applications seemed to give trouble with the video display, shifting the image to the left about one-third of the way and thus losing the left side

of the image. I believe this is a problem peculiar to my Tseng EVA/480 card and not endemic to EGA cards in general. IGC confirms that software supporting EGA graphics runs correctly. VM/386 arguably ranks among the highest of all the 386 operating environments in the compatibility department.

Late-Breaking News

Less than a week before the deadline for submission of this article, I received a pre-release copy of Version 1.1 of VM/386. This was partially in response to my complaints about slow disk access. Although there are a number of enhancements (a shared RAM-disk facility, support for Compaq DOS 3.31 large disk partitions, VGA support, etc.), little has been done to improve the basic problem of slow disk operations. I must therefore give VM/386 an overall grade of "C."

VM/386 excels in terms of ease of use, both while learning the system and in everyday operation. Compatibility with existing DOS applications is good, and processing overhead is not excessive.

On the minus side, the biggest drawback is the slow hard-disk routines. While this is offset to some extent through the optional use of a disk cache and a shared RAM disk, it is dismaying to see a state-of-the-art 80386 system with a 28-millisecond seek time drive respond like my first floppy-based 4.77-MHz PC. If IGC tackles this deficiency, VM/386 will be a winner. □

Charles Strom is a chemist by training, and has been working with computers for more than 10 years. He is a sysop on the GENie national time-sharing service.

Did you find this article particularly useful?
Circle number 8 on the reader service card.

Figure 1. The Virtual Machine Manager menu.

Product Information

VM/386 \$245

Intelligent Graphics Corporation

4800 Great America Parkway
Santa Clara, CA 95054
(408) 986-8373

System requirements:

- 80386 chip
- 1.2-MB, 5.25-inch, or 3.5-inch floppy drive
- Hard-disk drive
- DOS 3.0 or later
- 2 MB of RAM, minimum

by A.G.W. Cameron

Upgrading to a Sun 4

In the May/June and September/October 1987 issues of *Micro/Systems Journal*, I reported on my acquisition of a Sun 3/260 workstation and my early experiences with it. I have now upgraded to a Sun 4/260, which involved a swap of the CPU boards. That sounds like a trivial operation, but life is never simple. The Sun 4 series is characterized by Sun's new SPARC Architecture, and so I will begin with an account of what SPARC is all about.

SPARC Architecture

Sun Microsystems tells you that SPARC means Scalable Processor ARCHitecture, but ordinarily you are not told what that phrase really means. As defined by Sun, "scalable" describes a property of the smallest lines on a chip. If the pattern of these lines can easily be scaled to a different size, then it is called scalable. Sun says that its new CPU chips are scalable in this way, whereas most conventional architectures are sufficiently complicated that they do not scale well. If the pattern of lines on the chip can be significantly reduced in size, then generally the chip will run quite a bit faster. That property is at the heart of Sun's program to upgrade the performance of its workstations.

The most important aspect of the new architecture is that it is based on RISC (Reduced Instruction Set Computer) technology. There are 89 SPARC instructions, which is actually quite a large number, but much less than is used in many non-RISC computers. The instructions are all 32 bits in length, with a fixed format and word alignment. All instructions execute in a single cycle, except loads, stores, and floating-point operations. The more complex operations that would have been implemented as instructions in the past, and which would have taken many cycles to execute, are now performed by software.

There is a windowed register file that contains 120 registers. The presence of a large number of registers is one of the usual characteristics of RISC technology. The operations are carried out relatively simply. The instructions load the contents of a memory location to a register, store the contents of a register to memory, or operate on two registers or a register and a constant, with the result stored in a third register. The registers are organized in overlapping groups; each group has an input set, a local set, and an output set. The output registers of one group become the input registers of another group, and this defines the degree of overlap.

The design of the instruction set permits pipelined operation with four stages, so that several operations can be carried out simultaneously for efficient use. The efficiency is improved by arranging for branches to take effect after the execution of the next instruction, so that the next instruction can be fetched during the execution of the current instruction.

The SPARC CPU contains an Integer Unit (IU) that performs the basic operations, and it may or may not contain a Floating Point Unit (FPU). The current Sun implementation does not contain the FPU in the SPARC CPU, but rather it contains a floating point control unit. In principle, the FPU could be implemented externally with any of the floating point units commonly used with the Motorola 68000 series of chips or the Intel 386 chip. In practice, the Sun 4/260 uses the Weitek 1164/5 chip set that was previously available as an optional floating point accelerator (FPA) on the more expensive members of the Sun 3 series, and a little extra speed is attained with these chips in the SPARC design because they are implemented on the CPU board rather than on the optional FPA board. However, it is clear that Sun will have to move to other floating point options in the future in order to produce floating point performance at much increased speeds.

In order to reduce wait states for memory fetches, the 4/260 uses a 128-kilobyte cache on the CPU board. The 4/260 runs at 16.67 MHz, considerably slower than its 25-MHz Sun 3/260 ancestor (although the Weitek chips of the FPA that accompanied the 3/260 CPU board also ran at 16.67 MHz). Thus there is considerable room for speed improvement by simply increasing the clock speed of the CPU board. The Weitek chips currently run at 20 MHz on the Compaq 386/20, but probably they cannot run much faster than that.

Sun analyzes the performance of a given architecture in the following way. Let this performance be $P = S/IC$, where S is the clock speed, C is the average number of clock cycles per instruction, and I is the number of instructions in the benchmark. If I is in millions of instructions, then P will be in millions of instructions per second (mips). C is typically 1.3 to 1.7 for RISC machines and 4 to 10 for conventional architectures. RISC machines require typically 20 to 40 percent more instructions for a given algorithm, so for the same clock speed RISC machines will run two to five times faster than a conventional machine. Sun has thus obtained the numbers shown in Table 1, in which the SPARC performance is compared to that of the new Motorola 68030 and the Intel 80386, all running at the same 16.67-MHz clock speed. This is the basis for giving the 4/260 a 10 mips speed rating.

The Upgrade

My order for the Sun 4 upgrade over the Sun 3 contained three items: First, there was the new CPU board to be swapped for the old one. Second, with the Sun 4 series, FORTRAN is no longer bundled with the operating system, so it was necessary to order the new unbundled FORTRAN compiler, which has been implemented with a lot of VAX VMS compatible extensions. Third, I ordered one of the new 32-megabyte memory boards as an upgrade over the 8-megabyte board that I had (the delivery of this is now overdue).

The first item to arrive was the CPU board, together with the new operating system, UNIX for the SPARC. But I did not dare to do the upgrade without having the FORTRAN compiler, since that would have stranded us without the capability of doing any of our number-crunching. It was about a month before the Sun salesman was able to give me a tape with an advance copy of the new FORTRAN compiler. At the time he said that we should have the actual

C CODE FOR THE PC

source code, of course

	Bluestreak Plus Communications (two ports, programmer's interface, terminal emulation)	\$400
	CQL Query System (SQL retrievals plus windows)	\$325
	GraphiC 4.1 (high-resolution, DISSPLA-style scientific plots in color & hardcopy)	\$325
	Barcode Generator (specify Code 39 (alphanumeric), Interleaved 2 of 5 (numeric), or UPC)	\$300
NEW!	Vmem/C (virtual memory manager; least-recently used pager; dynamic expansion of swap file)	\$250
	Aspen Software PC Curses (System V compatible, extensive documentation)	\$250
	Greenleaf Data Windows (windows, menus, data entry, interactive form design)	\$250
	PforCe++ (COM, database, file, user interface, & CRT C++ classes among others)	\$345
	Vitamin C (MacWindows)	\$200
NEW!	TurboT _E X (TRIP certified; HP, PS, dot drivers; CM fonts; L _A T _E X)	\$170
	Essential resident C (TSRify C programs, DOS shared libraries)	\$165
	Essential C Utility Library (400 useful C functions)	\$160
	Essential Communications Library (C functions for RS-232-based communication systems)	\$160
	Greenleaf Communications Library (interrupt mode, modem control, XON-XOFF)	\$150
	Greenleaf Functions (296 useful C functions, all DOS services)	\$150
	OS/88 (U _{***} -like operating system, many tools, cross-development from MS-DOS)	\$150
	ME Version 2.0 (programmer's editor with C-like macro language by Magma Software)	\$140
	Turbo G Graphics Library (all popular adapters, hidden line removal)	\$135
	PC Curses Package (full System V, menu and data entry examples)	\$120
	CBTree (B+-tree ISAM driver, multiple variable-length keys)	\$115
	Minix Operating System (U _{***} -like operating system, includes manual)	\$105
	PC/IP (CMU/MIT TCP/IP implementation for PCs)	\$100
	B-Tree Library & ISAM Driver (file system utilities by Softfocus)	\$100
	The Profiler (program execution profile tool)	\$100
	Entelekon C Function Library (screen, graphics, keyboard, string, printer, etc.)	\$100
	Entelekon Power Windows (menus, overlays, messages, alarms, file handling, etc.)	\$100
	Wendin Operating System Construction Kit or PCNX, PCVMS O/S Shells	\$80
	C Windows Toolkit (pop-up, pull-down, spreadsheet, CGA/EGA/Hercules)	\$80
	Professional C Windows (windows and keyboard functions)	\$80
	JATE Async Terminal Emulator (includes file transfer and menu subsystem)	\$80
	MultiDOS Plus (DOS-based multitasking, intertask messaging, semaphores)	\$80
	WKS Library (C program interface to Lotus 1-2-3 program & files)	\$80
	Professional C Windows (lean & mean window and keyboard handler)	\$70
	Quincy (interactive C interpreter)	\$60
	EZ_ASM (assembly language macros bridging C and MASM)	\$60
	PTree (parse tree management)	\$60
	HELP! (pop-up help system builder)	\$50
	Multi-User BBS (chat, mail, menus, sysop displays; uses Galacticomm modem card)	\$50
	Make (macros, all languages, built-in rules)	\$50
	Vector-to-Raster Conversion (stroke letters & Tektronix 4010 codes to bitmaps)	\$50
	Coder's Prolog (inference engine for use with C programs)	\$45
	C-Help (pop-up help for C programmers ... add your own notes)	\$40
	Biggerstaff's System Tools (multi-tasking window manager kit)	\$40
	PC-XINU (Comer's XINU operating system for PC)	\$35
	CLIPS (rule-based expert system generator, Version 4.1)	\$35
	TELE Kernel or TELE Windows (Ken Berry's multi-tasking kernel & window package)	\$30
	Clisp (Lisp interpreter with extensive internals documentation)	\$30
	Translate Rules to C (YACC-like function generator for rule-based systems)	\$30
	6-Pack of Editors (six public domain editors for use, study & hacking)	\$30
NEW!	Crunch Pack (a dozen file compression & expansion programs)	\$30
	ICON (string and list processing language, Version 6 and update)	\$25
	LEX (lexical analyzer generator)	\$25
	Bison & PREP (YACC workalike parser generator & attribute grammar preprocessor)	\$25
	AutoTrace (program tracer and memory trasher catcher)	\$25
	C Compiler Torture Test (checks a C compiler against K & R)	\$20
	Benchmark Package (C compiler, PC hardware, and Unix system)	\$20
	TN3270 (remote login to IBM VM/CMS as a 3270 terminal on a 3274 controller)	\$20
	A68 (68000 cross-assembler)	\$20
	List-Pac (C functions for lists, stacks, and queues)	\$20
	XLT Macro Processor (general purpose text translator)	\$20
	Data	
	WordCruncher (text retrieval & document analysis program)	\$275
	DNA Sequences (GenBank 52.0 including fast similarity search program)	\$150
	Protein Sequences (5,415 sequences, 1,302,966 residuals, with similarity search program)	\$60
	Webster's Second Dictionary (234,932 words)	\$60
	U. S. Cities (names & longitude/latitude of 32,000 U.S. cities and 6,000 state boundary points)	\$35
	The World Digitized (100,000 longitude/latitude of world country boundaries)	\$30
	KST Fonts (13,200 characters in 139 mixed fonts: specify T _E X or bitmap format)	\$30
	USNO Floppy Almanac (high-precision moon, sun, planet & star positions)	\$20
	NBS Hershey Fonts (1,377 stroke characters in 14 fonts)	\$15
	U. S. Map (15,701 points of state boundaries)	\$15

The Austin Code Works

11100 Leafwood Lane

Austin, Texas 78750-3409 USA

acw!info@uunet.uu.net

CIRCLE 62 ON READER SERVICE CARD

Voice: (512) 258-0785

BBS: (512) 258-8831

FidoNet: 1:382/12

Free shipping on prepaid orders

For delivery in Texas add 7%

MasterCard/VISA

Table 1. Processor Performance

CPU	I	C	S	P
Motorola 68030	1.0	5.2	16.67	3.21
Intel 80386	1.1	4.4	16.67	3.44
SPARC	1.2	1.3	16.67	10.69

Table 2. Comparative Benchmarks of the Sun 4/260 and Other Workstations

Computer	S-whet kwhet/s	D-whet kwhet/s	REMAX sec.	SVD sec.	Roots sec.
Sun 4/260	5219	3281	0.48	1.36	0.45
Sun 3/260 FPA	3561	2373	0.65	2.21	0.68
VAX 11/780 FPA	1191	734	3.0	n/a	n/a
Definicon 780+/4	1157	1051	2.41	15.22	4.45
VAX 11/785 FPA	1800	1157	1.3	7.9	0.399
VAX 8650 FPA	6100	3895	0.53	2.2	0.130
IBM 3081D	5850	5680	0.30	n/a	n/a

WHY SETTLE FOR THIS?...

$$\int_{-\infty}^{\infty} \frac{y \cdot z}{x - tY} dt = \sum_{i=1}^n \frac{(x^4 - i)^3}{\sqrt{Z^3 + y^{10}}} \frac{5}{Z - X^5 Y}$$

...WHEN WHAT YOU REALLY WANT IS THIS.

$$\int_{-\infty}^{\infty} \frac{Xy \cdot Z}{X - tY} dt = \sum_{i=1}^n \left(\frac{(x^4 - i)^3}{\sqrt{Z^3 + y^{10}}} \frac{5}{Z - X^5 Y} \right)$$

TRY THIS:**PERSONAL****INC**

Quality work should look high-quality. What better reason to try **PC_{TEX}**®—the full implementation of Prof. D. Knuth's revolutionary **TEX** formatting/typesetting program.

PC_{TEX} offers professional typesetting capabilities & advantages to PC users. It gives you control—of format, type, symbols, quality—for complex mathematical &

To order or for information, call:

415/388-8853

or write: Personal **TEX**, Inc.
12 Madrona Avenue
Mill Valley, CA 94941 USA

PC_{TEX} is a registered TM of Personal **TEX**, Inc.
TEX is an American Mathematical Society TM.
Manufacturers' product names are their TMs.

engineering material, statistical tables or straight matter.

So whether you're writing the next starshot manual or a thesis on relativity, you get camera/publisher-ready manuscripts to be proud of, quick & simple. Don't settle for less.

From Personal **TEX**, Inc., starting at \$249; VISA/MC welcome. Satisfaction guaranteed.

PC_{TEX} FORMATTING/TYPESSETTING SYSTEM • **FINE TYPESET QUALITY** from dot matrix or laser printers, or phototypesetters. • **A COMPLETE PRODUCT.** Includes • our specially written **PC_{TEX}** Manual that lets you use **TEX** immediately • custom 'macro package' formats for letters, manuals, technical documents, etc. • the **La_{TEX}** document preparation system (with user's manual) macro package for article, book, report preparation • **AMS-_{TEX}**, developed by the American Mathematical Society for professional mathematical typesetting. • **OUTPUT DEVICE DRIVERS** available for Epson FX, LQ • Toshiba • HP LaserJet Series • Apple LaserWriter • Screen preview, with EGA, VGA or Hercules card. • **REQUIRES:** IBM PC/XT, AT or compatible, DOS 2.0 or higher & 512K RAM; hard disk for printer drivers & fonts.

This ad is typeset & composed using: PC_{TEX}, Bitstream® fonts & laser printer. Logotype & black backgrounds done photographically.

shipment of my FORTRAN order in two or three weeks, but more than that time has now gone by and it has still not arrived. I do not know whether the FORTRAN version that I have is a beta version that was lying around the sales office, or whether it is the version that I am supposed to get. I also had to wait until a new version of the UNIX for the Sun 3 became available that would run compatibly with the SPARC operating system, so my upgrade would not cut off one of my colleagues who is operating his Sun 3/110 using my hard disks as a file server.

With FORTRAN in hand, and having newly backed up our files, I swapped the new CPU board for the old one, which turned out to be a relatively painless operation, not much more involved than swapping boards in a PC. I also took out my old FPA and turned it over to my colleague to install on his Sun 3/110. The installation of the new operating system went smoothly, and shortly we were back on the air. However, two problems then surfaced. The first was that so far we have not been able to interface a modem to the new operating system, despite having tried to set it up in the same way that worked previously with the Sun 3. The operating system tried to generate lots of spurious logins. That remains to be fixed. The other problem is that Sun has so far not released the Transcript software for the SPARC operating system so I cannot run my LaserWriter from the Sun 4, a considerable nuisance that required me to relocate the LaserWriter to a PC. I am told that I will get an alpha copy of the new Transcript software about May. We will see.

We normally spend most of our CPU cycles running a three-dimensional hydrodynamics code that runs 24 hours per day. My main motivation in trying to get faster machines is to be able to do more of these calculations in a smaller amount of time. I was expecting to approximately double the speed on this program, since the Sun 3/260 was a 0.5 mflops (millions of floating point operations per second) machine as determined from the LINPACK benchmark, and the Sun 4/260 has been described as a 1.1 mflops machine. But we only found a 15 percent increase in speed. At the moment we do not know how much of this discrepancy to attribute to the effects of a new architecture and how much to the effects of an immature FORTRAN compiler.

It may, therefore, be of some

interest to see the results of a number of other floating point benchmarks that I have run. Table 2 shows the results of five floating-point intensive benchmarks that were first used to compare a large number of machines by Avram Tetewsky and Dan Feenberg (see *Micro/Systems Journal*, September/October 1987). The Sun 4 results are the new benchmarks; the Sun 3 and Definicon measurements were also made by me; and the other comparison numbers came from Tetewsky and Feenberg. It may be seen that there is a significant variation in the ratio of speeds between the Sun 4 and the Sun 3; this ratio varies from 1.4 to 1.6. This indicates that there are definitely effects due to architectural differences in these benchmark results, although they are not major. I also found that the benchmark performances varied by factors of 1.1 to 1.6 between compiling with optimization and compiling without it. However, we have not noticed any performance improvement in using optimization on the hydrodynamics code as compared to no optimization.

My present hypothesis is that the poor performance of the hydrodynamics code compared to these other benchmarks is due to the effects of the 128-kilobyte cache. The benchmark programs are forced to have large arrays, but only a small part of these arrays is actually addressed. Thus the data and code that is actually addressed will fit into the cache. However, the amount of data and code that is addressed in the hydrodynamics program is much larger than the cache, and the data is more or less randomly addressed, so that in the hydrodynamics program there will be a much higher fraction of memory loads into the cache than for the benchmark programs. Normal FORTRAN optimization would not change this behavior.

The Future of SPARC

For what it is worth, I will repeat a rumor that was going around in early March about the next major Sun workstation, whose announcement is expected this month. It is rumored to run at 25 mips, with 10 mflops double precision, and with 20 to 30 mflops single precision. This is an enormous performance jump. But it may not be an unreasonable one.

One of the SPARC licensing deals that Sun Microsystems has made is with Cypress Semiconductor, which has already announced that much faster versions of the SPARC chips will become available this spring. Twenty-five mips is about in accord

with the faster speed of the Cypress chips. Sun has recently announced that the TAAC1 Applications Accelerator, a graphics-intensive insert board that can also be used for fast number-crunching, has a floating point performance comparable to the above numbers. This insert board derives its speed from new floating point chips made by Texas Instruments, which, unlike the Weitek chips and like the Intel 8087 family, implement the full IEEE standard for floating point operations (i.e., they do square roots, exponentiation, logarithms, and some trigonometric functions). The TAAC1 was developed by a small start-up in North Carolina which was

recently acquired by Sun.

I look forward to the next upgrade.

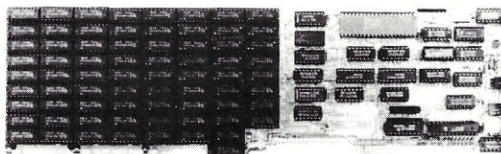
A.G.W. Cameron is Professor of Astronomy at the Harvard-Smithsonian Center for Astrophysics.

Did you find this article particularly useful?
Circle number 9 on the reader service card.

Product Information

Sun Microsystems Inc.
2550 Garcia Avenue
Mountain View, CA 94043
(415) 960-1300

GET A FLAME



The Blue Flame II is the latest in our line of very high-performance disk emulators for PC's, XT's, AT's, '386's, and all clones. It's extremely fast: 800Kbytes per second transfer rate, ten times faster than hard disks. Even faster than IBM's VDISK program! And big: Up to 8 megabytes per board, 32 megabytes per logical drive. Much bigger than extended or expanded memory. It doesn't waste any of your computer's memory address space for storage. And the Blue Flame II is reliable: With no moving parts, it can be accessed continuously for years with no failures. Don't try this at home with your hard disk!

Not just another RAMdisk, the Blue Flame II has an external AC-powered battery-backup option: Data isn't lost when the computer is turned off. And "Reset" isn't a dirty word anymore. Even during a blackout, the battery maintains data for 10 hours.

The Blue Flame II is available fully-populated, with 8 megabytes, for \$2095. 4 megabytes for \$1195. 2 megabytes for \$795. Battery Backup option costs \$135. Call us for information on our SemiDisk products for S-100, and Epson QX-10/QX-16.

If you want greater software speed, improved data security, increased hardware reliability, get a Flame. If you need the hottest disk performance possible, get a Flame. A Blue Flame II SemiDisk.

SemiDisk Systems, Inc.

P.O. Box GG
Beaverton, OR 97075
(503) 626-3104

CIRCLE 47 ON READER SERVICE CARD

by Bob Blacher

Zmodem—Why a New File Transfer Protocol?

Zmodem? What the heck is that? I've got Xmodem, Xmodem CRC, and even something called Ymodem. But what's Zmodem?

I've seen quite a few callers to my bulletin board system (BBS) pause for a long time when presented with the list of file transfer protocols that are offered. On my board, the file transfer protocols available are Zmodem, "true" Ymodem, and Xmodem-1K. The old standbys, Xmodem & Xmodem CRC, are conspicuously absent. Clearly, something has changed, but this change has received little attention outside of the BBS community and has been missed by most publications, even those dedicated to serious computer users. It's time to remedy that situation.

It makes little sense to talk about all the software that's available on BBS systems without first discussing how to transfer copies of the software from the BBS' computer to yours as fast and reliably as possible. So, let me start my contribution to this column by talking about file transfer protocols, especially Zmodem and its implementation in Chuck Forsberg's DSZ program.

Transmission Speed Outpaces the Xmodem

File transfers sure were simple in the good old days, huh? The RCP/M systems supported Xmodem, period. Ward Christensen's Xmodem served us all well for quite a few years. Yes, improvements were made, such as using CRC error-checking instead of the original checksum method (consult your favorite expert who speaks polynomials to determine the differences—I don't have room here). But, in essence, Xmodem was Xmodem, and every CP/M modem program had it and that's what counted.

The DOS world actually regressed at first, going back to Xmodem checksum and only slowly adopting Xmodem CRC. The CP/Mers sneered,

and the DOS folks caught up. Xmodem remained the standard in both worlds.

Within the last couple of years, protocols have become decidedly more complicated. The communications environment has changed. When Christensen first developed the Xmodem protocol, modems that transmitted 300 bits per second (bps) were the fastest around. The advent of 1200 bps modems didn't really cause any problems for the Xmodem protocol either. Right now, 2400 bps modems are the norm (over 50 percent of the callers to my BBS call at 2400 bps), and 9600 bps modems are slowly gaining acceptance, but are hampered by incompatibilities that hopefully will be resolved later this year.

Faster modems push the current voice telephone lines to their limits. Error detection and recovery become ever more critical. The phone lines we all use weren't designed for this kind of data transmission, but we're stuck with them and file transfer protocols have to deal with this reality.

Packet Switching—Death Knoll for Xmodem

Then, there are packet-switching networks. The best known is GTE's PC Pursuit that offers unlimited modem calls during off-peak hours for a flat, monthly fee. PC Pursuit, and services like it, have had a dramatic effect on BBS systems, leading many boards to be truly national with callers from every part of this country. But, a price has to be paid—speed. First, PC Pursuit still offers only 1200 bps service in most parts of the country. And, packet-switching murders Xmodem transfers. Efficiency drops dramatically, and reliability goes down as well. Some BBS system operators (sysops) started muttering about banning PC Pursuit callers from their boards altogether. It was time for some new file transfer protocols to be developed.

The Xmodem protocol is pretty easy to understand when it's reduced to its basics. A file is sent in 128-byte

packets. Each packet also contains a few bytes of identifying information in a header and the checksum or CRC value of the packet. The sender builds the packet, sends it along to the receiver and *stops!* The sender must wait for the receiver to do its own checksum or CRC calculation on the packet, confirm that it matches what the sender said it would be, and transmit an acknowledgment back to the sender. Only when the sender gets this acknowledgment can the next packet be sent. (If the receiver says that an error has occurred, then the packet is re-sent until a positive acknowledgment is received.)

The details of all this aren't critical for present purposes. The key is the start/stop nature of the protocol—send a packet, wait for an acknowledgment, send another packet, wait again, and so on. That's fine on a fast, clean telephone line. The waiting period between packets is short and the efficiency of the transfer is acceptable.

That same transfer over a heavily loaded packet network is painful to watch. Don't hold me to this number, but assume that the network imposes a two-second delay on turnarounds and look what just happened to Xmodem. Send a measly 128 bytes of data, wait two seconds!, send another 128 bytes, wait two more seconds!, and so on. No wonder the sysops were pulling their hair out and grumbling about packet network callers.

Packet networks aren't the only culprits. In this world of alternative long distance carriers, phone calls are bouncing off satellites, hopping over microwave relays and the like. In addition, the lower cost 9600 bps modems that are currently used by BBS systems only simulate full-duplex transmission at 9600 bps and up. These modems impose their own delays on data turnarounds. While the packet nets may impose the worst delays, delays between packets were reducing the file transfer efficiency of many callers.

Enter Ymodem

If the problem with Xmodem is that we're getting delays with every 128 bytes of data, there would seem to be a simple solution—increase the packet size. Ymodem is born. The term "Ymodem" was coined by Ward Christensen in 1985, but the protocol is actually based on work done by Chuck Forsberg in his CP/M, and later DOS modem programs called YAM. Undoubtedly, I'm going to be struck by a bolt of lightning for putting it this way, but reduced to

essentials, Ymodem is Xmodem with 1024 byte (1K) packets instead of 128 byte packets.¹ The result is the sender only has to wait for acknowledgments from the receiver one-eighth as often and everything should go along much more smoothly (so to speak).

Until an error occurs, that is. With Xmodem, retransmissions due to an error meant sending the same 128 bytes of data again. With Ymodem, 1K has to be retransmitted. The gains we expected by having only one-eighth the number of packets are going to disappear pretty quickly on a noisy line when each error means retransmitting eight times as much data. On a clean line, Ymodem is an excellent, fast protocol, but it's actually going to be less efficient than Xmodem if a significant number of packets must be sent more than once because of errors.

X, Y, and Introducing Z

Enter Zmodem; exit start/stop protocols. Zmodem was developed by Chuck Forsberg of Omen Technology, Inc., under a contract from Telenet, a large, timesharing network. The Zmodem protocol description is public domain. Developers of communications programs who wish to add Zmodem may do so freely—no license is required. UNIX and VMS C source for Zmodem, written by Forsberg, is also in the public domain.

Zmodem operates on the "Don't call me, I'll call you" method of acknowledgments between the sender and the receiver. In brief, the sender doesn't wait for an acknowledgment after each packet of data. It just keeps sending data in what looks to be a continuous stream. If the receiver is silent, then the sender assumes everything is okay and keeps on sending. It's only when the receiver actually informs the sender that a mistake has occurred that the sender goes back to the point of error and

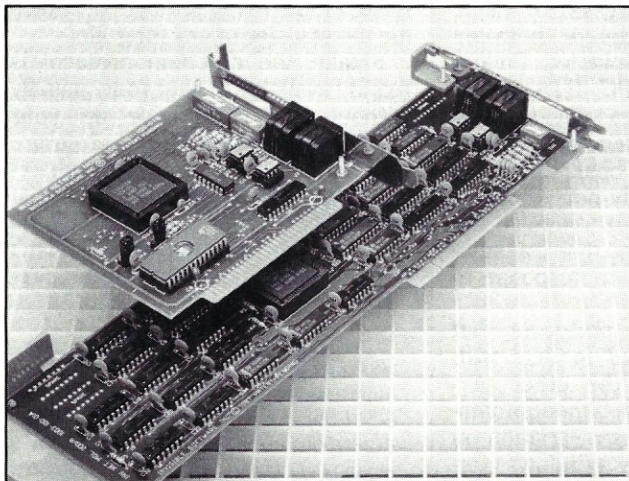
begins retransmission again from that point. The key is that the receiver not only tells the sender that an error occurred, but also its position in the file. The sender is capable of repositioning to the point of the error, even if several packets have been sent since that time.

It's actually easier to understand what's going on if you watch the lights on an external modem. With either Xmodem or Ymodem, the Send Data (SD) light on the sender's modem goes on as each 128-byte or 1K packet is sent. There's a pause, then a brief flash of the Receive Data

(RD) light as the acknowledgment comes back, and then the SD light comes on again as the next packet is sent.

A Zmodem transfer looks completely different. The SD light is on continuously and nothing comes back from the receiver unless an error occurs. If there is an error, you'll see a brief flash of the RD light, the sender will reposition itself to the place in the file where the error was detected, and then start sending the file again in a stream.

The issue of how much of the file has to be retransmitted if an error



Best value in a complete LAN New! Invisible Network.

Best value. New Invisible Network™ from Invisible Software is a *real* LAN. It transfers data at 1.8Mbps. In one high performance package, it combines power, reliability, ease of installation, and low cost. We have included all the hardware and software. And backed it with a lifetime limited warranty plus a 30-day money-back guarantee.

Broad compatibility. Invisible Network includes the NET/30™ Network Operating System at no additional expense. Invisible Network is also fully compatible with the IBM NETBIOS standard, the IBM-PC LAN Program™, and Novell Advanced Network™. So you can run all the most popular network applications.

Installs simply. You easily install the LAN yourself by inserting an interface card (with built-in modular jack) into each network computer. It's that simple.

	Model 100	Model 200	Model 200/A
Compatibility	IBM PC/XT/AT	IBM PC/XT/AT	IBM PS/2 Micro Channel
Speed	0.7 Mbps	1.8 Mbps	1.8 Mbps
Price	\$199.00	\$299.00	\$399.00

For more information call (415) 221-0916 or write.

CALL ADVERTISER DIRECTLY

"Good performance under load, elegant design, low price, simple cabling requirements, and broad compatibility."

—PC Week

"The Invisible Network is a darn good deal."

—PC Magazine

"A joy to use. I highly recommend it."

—Data Based Advisor

Invisible Software, Inc.
481 47th Ave.
San Francisco, CA 94121
(415) 221-0916

¹ "True" Ymodem refers to a batch (multiple file) protocol that includes a special Block 0 at the start of the transfer with filename, file length, and modification date information. If a so-called Ymodem implementation is nothing more than Xmodem CRC with 1K packets, it is more accurately referred to as Xmodem-1K. The issue is not purely theoretical since it has led to incompatibilities with some communications programs. If your communications program claims to support both Ymodem Batch and Ymodem, the latter is probably Xmodem-1K. Many BBS systems only support Xmodem-1K, often improperly labeling it Ymodem.

occurs is handled quite ingeniously by Zmodem. Packet size is dynamically adjusted during the transfer based on the number of errors. The "typical" Zmodem packet size is 1K, the same as Ymodem, but as errors mount, the packet size will be dropped to 512 bytes, then 256 bytes, then 128 bytes, and so on, to a minimum packet size of 32 bytes. This "downshift" in packet sizes improves efficiency on poor lines. If line conditions improve during a transfer, the Zmodem implementations can turn around and upshift packet sizes again, all the way back to 1K packets.

It's really quite impressive to watch. The problem of delays between packets is all but eliminated. Callers on networks like PC Pursuit get roughly the same transmission rates as do callers on "regular" phones lines (albeit, at 1200 bps). Efficiency is good even on fairly noisy lines as packet sizes adjust to minimize the amount of data that must be transmitted more than once in response to an error signal.

There are other protocols that have been developed to deal with packet delays and the other perceived deficiencies in Xmodem family transfers. There is a variation on Kermit called Sliding Windows Kermit, an Xmodem variant called WXmodem (Windowed Xmodem), a protocol developed by Thom Henderson of ARChive fame called SeaLink, and another by Paul Meiners called MegaLink. Each of these has its fans, and each performs well in certain situations. In my own experience, none has turned in as consistently reliable and efficient perform-

ance as Zmodem.

And, there's a bonus in Zmodem that warms the heart of every sysop that runs a busy system. There's nothing more frustrating for either a caller to a BBS, or for the sysop, than watching a transfer abort well into a long file. Precious time and, in the case of a long distance call, money have been wasted. Zmodem has a unique "crash-recovery" feature. With Zmodem, even if you drop carrier 200K into a 250K file, you can call back and you'll only have to download the remaining 50K of the file. That feature alone would tend to make Zmodem my protocol of choice for the transfer of any long file.

Finally, the Zmodem protocol is readily available to everyone, usually without having to give up your favorite communications program. Chuck Forsberg has created a shareware *protocol driver* called DSZ that supports Zmodem, Ymodem, and the Xmodem-family protocols.² DSZ isn't a full-fledged communications pro-

² The DSZ program for Zmodem file transfers is available for download from almost any BBS. It is shareware, with a \$20 license fee requested after initial evaluation. Copies may be ordered by sending a check for \$20 to: Omen Technology Inc., P.O. Box 4681, Portland, OR 97208. The DSZ diskette includes complete specifications for both the Ymodem and Zmodem protocols. These protocol documents are public domain and are available from many BBS systems, typically in a file called YZMODEM.ARC. Omen Technology runs its own BBS (Professional-YAM in host mode) at (503) 621-3746 [1200/2400/19200 (Telebit PEP)].

gram. (Forsberg's excellent, if somewhat complex, Professional-YAM and its shareware subset ZCOMM are complete communications packages supporting Zmodem and every other imaginable file transfer protocol.) Rather, DSZ does one thing and does it wonderfully—it transfers files. Most BBS systems that support Zmodem are, in fact, using the DSZ program to handle Zmodem transfers. Callers to BBS systems can do the same.

Almost all of the popular communications programs that are available on BBS systems, including ProComm Plus, Qmodem SST, PibTerm and others, support external protocol drivers, such as DSZ, and make it quite easy to use Zmodem. In fact, any communications program that allows you to run DOS commands, which includes most commercial communications programs, will allow you to use DSZ for Zmodem transfers.

Next time you're on a BBS, I hope you'll make DSZ your first download. In future columns, we'll talk about other excellent programs that are available from BBS systems, and maybe you'll even begin using Zmodem to get copies of these programs for yourself with maximum speed and reliability.

Bob Blacher operates the popular Computer Connections PCBoard bulletin board system in Washington, D.C. It is quite busy, but those willing to place their modems on redial may call (202) 547-2008. In his other life, Bob is an attorney.

Did you find this article particularly useful? Circle number 10 on the reader service card.

PERISCOPE™

POWER

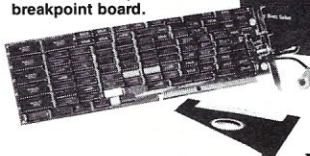
...Keeps you going full steam ahead when other debuggers let you down. With four models to pick from, you'll find a Periscope that has just the power you need.

... Start with the model that fits your current needs. If you need more horsepower, upgrade for the difference in price plus \$10! And don't worry about having a lot more to learn ... Even when you move to the most powerful model, Periscope III, an extra dozen commands are all that's involved.

Periscope's software is solid, comprehensive, and flexible. It helps you debug just about any kind of program you can write ... thoroughly and efficiently. Periscope's hardware adds the power to solve the really tough debugging problems.

Periscope requires an IBM PC, XT, AT, or close compatible (Periscope III requires hardware as well as software compatibility); DOS 2.0 or later; 64K available memory; one disk drive; an 80-column monitor.

Top-of-the-line Periscope III with real-time, hardware breakpoint board.



Periscope I includes a half-length board with 56K of write-protected RAM; break-out switch; software and manual for \$345.

Periscope II includes break-out switch; software and manual for \$175.

Periscope II-X includes software and manual (no hardware) for \$145.

Periscope III includes a full-length board with 64K of write-protected RAM, hardware breakpoints and real-time trace buffer; break-out switch; software and manual. Periscope III for machines running up to 8 MHz is \$995; for machines running up to 10 MHz, \$1095.

Call Toll-Free for free information or to order your Periscope today!

MAJOR CREDIT CARDS ACCEPTED.

800-722-7006

The
PERISCOPE
Company, Inc.

1197 PEACHTREE ST.
PLAZA LEVEL
ATLANTA, GA 30361
404/875-8080

CIRCLE 66 ON READER SERVICE CARD

me anything about IND\$FILE (how to invoke it, what it expects to see, etc.). I know that the IBM and IRMA PC emulation packages both use it, and in the column Mr. Cherry mentioned how to use it. My question is, "How?"

One of the projects in which I am currently involved concerns file transfers between our IBM 3090/600 mainframe and our PCs. We currently have a variety of 3278 emulation cards for our PCs (IRMA, Forte, Quadram/Forte, and IBM) and I am attempting to come up with a program for universal file transfers of data between a PC and our mainframe.

We own Tempus-Link and use it quite a bit, but we would like to be able to transfer certain data files between the mainframe and the PC without going through Tempus-Link's multitude of commands and intermediate steps. For the specific project in which I am involved, we feel that Tempus-Link would be too cumbersome to be of any great use, and we don't want to have to purchase a copy of the IRMA file transfer program for each user.

Essentially, the file transfer program that I am attempting to write would not have to be generic (although it would be great if I could make it generic). Right now, I am just trying to come up with a program to work with a system my area is writing. Any information on IND\$FILE would be greatly appreciated.

Mark E. Davidson
Provident Life & Accident
Chattanooga, Tenn.

From the Author:

There is no specific program from IBM called IND\$FILE. The program is actually called the IBM 3270 PC File Transfer Program. When it is installed on an IBM mainframe, a module named IND\$FILE is created on the host. Consequently, the name IND\$FILE has become slang for the IBM 3270 File Transfer Program.

The program itself comes in two varieties: one for use with MVS/TSO, and the other for use with VM/CMS. The product number for the TSO version of the product is 5665-311.

In that IND\$FILE is generally already installed on most IBM mainframes (especially if the company has used 3270 PCs), a large number of companies have made their file transfer programs support the

IND\$FILE module. For example, DCA also supplies PC-based file transfer software that will work with the IND\$FILE host module. It is called IRMA FT/3270 (as opposed to FT/TSO, that works with a proprietary IRMA host program under CMS). Eicon also supports IND\$FILE transfer from their SDLC and X.25 gateways.

The actual instructions to upload a file from a PC to TCO would differ based on the PC hardware and software being used. Basically, the instructions would be something like this:

1. Log on to TSO in terminal emulation mode.
2. Get to the TSO Ready prompt (do not access SPF).
3. Hot-key to DOS on the PC.
4. At the DOS prompt, type:

- *PCfile.nme* is the name of the DOS file on the PC.
- *TSO.file.name* is the name of the TSO data set. Note that a specific member for a partitioned data set can be included.
- *ASCII* is an optional parameter that translates ASCII characters to EBCDIC characters.
- *CRLF* is an optional parameter that edits the carriage returns and line feeds.

For example, to send the DOS file STUFF.TXT from the C: source drive on the PC to the TSO data set TSUSMY.LAN.TEXT(STUFF), you would type the following:

```
SEND C:STUFF.TXT  
TSUSMY.LAN.TEXT (STUFF) ASCII CRLF
```

Note that if you are allocating the TSO file during the upload, there are specific data set allocation parameters that can be specified on the command line.

To download a file from TSO to DOS, the command would be:

```
RECEIVE PCfile.nme TSO.file.name  
ASCII CRLF
```

Again, it is necessary to carefully review the specific documentation with your particular host link and software.

The commands for file transfer to and from CMS are similar, the main difference being the CMS data set naming convention differs from the TSO data set naming convention.

Michael Cherry
Denver, Colo.

Introducing

NANODISK

"Disk Cache for the IBM PC"

Make your floppy drive and hard disk run close to RAM disk speeds. Dramatic speed improvement for most programs. Supports cache of any size in main or expanded memory.

Requires IBM PC/XT/AT or true clone.

only **\$29.95**

MultiDos Plus

"multitasking for the IBM-PC."

Ideal for developing applications in process control, data acquisition, communications, and other areas. Check these features which make **MultiDos Plus** an unbeatable value.

- Run up to 32 DOS programs concurrently.
- Operator commands to load/run programs, change priority, check program status, abort/suspend/resume programs.
- Programmatic interface via INT 15H for the following:
 - * Intertask message communication.
 - * Task control by means of semaphores.
 - * 256 priority levels.
 - * Suspend task for specified interval.
 - * Spawn and terminate external and internal tasks.
 - * Disable/enable multitasking.
 - * and more!

Requires IBM PC/XT/AT or true clone, and enough memory to hold **MultiDos Plus** (48 KB) and all your application programs.

\$24.95 or **\$99.95**
With source code
(Written in Lattice C
and Microsoft Assembler.)

Outside USA add \$5.00 shipping and handling.
Visa and Mastercard orders only call toll-free: 1-800-872-4566, ext. 350., or send check or money order (Drawn on U.S. Bank Only) to:

NANOSOFT

13 Westfield Rd, Natick, MA 01760
MA orders add 5% sales tax.

CALL ADVERTISER DIRECTLY

New Products

New Software Products

386/ASM Now Supports 80376

Phar Lap Software is now offering support for Intel's 80376 embedded processor in its 386/ASM/LinkLoc software package. The package provides a set of development tools for the 80376/80386 microprocessor family, and includes 386/ASM, a 32-bit mode assembler, and a linker-plus-locator utility. 386/ASM is compatible with the Microsoft MASM assembler. LinkLoc is used by the programmer to link 80376 ROM-based high-level language software and produces executable image files that can be downloaded to in-circuit emulators, ROM-based debuggers, or other targets. Command switches allow the programmer to select control numbers, the attributes and size of segments, and where programs are located in memory.

The 386/ASM/LinkLoc package includes the 386/ASM and LinkLoc software, a user's manual, and sample programs. The PC version is priced at \$695. For more information, contact **Phar Lap Software Inc.**, 60 Aberdeen Avenue, Cambridge, MA 02138; (617) 661-1510.

Circle number 11 on the reader service card.

Security is a Secret(Disk)

SecretDisk II is Lattice Inc.'s latest data security and encryption package. SecretDisk II is designed to create new logical disk drives. By creating hidden system files on floppy or hard disks, all data can be automatically encrypted. Protection can be turned on and off at any time without rebooting. Pass-

word-protected access to protected files is provided through a DOS device driver. "Secret disks" also can be built inside other "secret disks" to provide multiple security levels, with encryption methods varying at each level. To the user, protected disks are the same as any other disk drive. When the drives are "unlocked," files can be easily copied in and out of the protected area. Databases, spreadsheets, and other application programs recognize protected drives and will read and write to "secret" disk drives as they will to drives C: and D:.

SecretDisk II is offered at \$79 for MS-DOS systems, and versions for OS/2 and UNIX are promised in the near future. For more information, contact **Lattice Inc.**, 2500 South Highland Avenue, Lombard, IL 60148; (312) 916-1600.

Circle number 12 on the reader service card.

Sherlock Hunts C Coding Clues

Edward K. Ream has begun shipping copies of Sherlock, a product developed to provide programmers with insight into the inner workings of C programs. Sherlock acts as a magnifying window into a program using macros and support routines. The macros define named code boxes containing tracing and debugging code supplied by the programmer. The boxes conceal the code until needed, and special command-line arguments enable the boxes. This makes it easier to code, test, and maintain complex programs. Sherlock works with Microsoft, Turbo C, and other C compilers. It also comes with three utility programs: the SPP program that in-

serts macros; the SDEL program that removes Sherlock macros; and the SDIF to compare files with Sherlock macros to those without.

Sherlock is \$125. For more information, contact **Edward K. Ream**, 1617 Monroe Street, Madison, WI 53711; (608) 257-0802.

Circle number 13 on the reader service card.

Source Code Ready for Assembly

Abraxas Software has released PCYACC Version 2.0, a program generator capable of creating C or Pascal source code for building assemblers, compilers, calculators, type-setting languages, and other functions. PCYACC is designed to generate C or Pascal code optimized for Microsoft, Borland, or Lattice compilers. Runtime library and example sources are provided as applications skeletons. PCYACC is input-compatible with UNIX YACC when used with C grammar descriptions.

PCYACC is available for \$139 for the personal version or \$395 for the professional version from **Abraxas Software**, 7033 S.W. Macadam Avenue, Portland, OR 97219; (503) 244-5253.

Circle number 14 on the reader service card.

89 Math Functions In One Toolbox

Silver State Software has begun shipping the C Math Toolbox, a set of 89 math functions for Microsoft QuickC and C optimizing compilers. Included are statistical, complex number, image analysis, and numerical analysis, including Fast Fourier Transforms. Each function in the Toolbox is delivered in three separate forms: as source code, in an optimized linkable library, and in a CodeView linkable library. The source code files are in correct form for use with QuickC, either merged, included, or as .MAK file

components. Included in the package is a 200-page manual, 89 function source files, 89 example programs, 50 .MAK files, a CodeView linkable library, and an optimized linkable library.

The C Math Toolbox is available for \$89. For more information, contact **Silver State Software**, 1000 E. William Street, Suite 100, Carson City, NV 89701; (702) 885-8922.

Circle number 15 on the reader service card.

Software Protocols Make Migration Easy

Network Software Associates has released Version 4.0 of AdaptSNA, a PC-to-host communications software package for IBM PCs, XT's, AT's, and PS/2's. The new edition of AdaptSNA implements a full range of SNA protocols, including 3270, Logical Unit 6.2 Advanced Program-to-Program Communications (LU6.2/A PPC), 3770 Remote Job Entry (3770/RJE), and Logical Unit type 0 (LU0). AdaptSNA is hardware- and link-independent, so the same software can be used over any physical link. Users have a cost-effective migration path so they can change physical transmission links without losing applications interfaces or procedures. For example, APPC cooperative processing can be implemented using the PC's existing serial port (COM1 or COM2) and an asynch modem, then gradually migrating to an SDLC or coax connection without rewriting applications.

Included in the software family are AdaptSNA 3270 for interactive host communications (\$585), AdaptSNA APPC for cooperative processing applications (\$285), AdaptSNA RJE (\$785) for batch file transfers, and AdaptSNA LU0 for program-to-program communications. For more information, contact **Network Software Associates Inc.**, 22982 Mill Creek, Laguna Hills, CA 92653; (714) 768-

4013.

Circle number 16 on the reader service card.

New Hardware Products

Comm Board Provides PC-to-Host Link

Network Software Associates Inc. has introduced a SynchronousDataLink Control (SDLC) communications board, the AdaptCom model C for IBM PCs, XTs, ATs, Model 30s, and compatibles. The half-slot board is designed to be used with the company's AdaptSNA software for PC-to-host and PC-to-PC communications at rates up to 19,200 Kbps. AdaptCom provides hardware/software communications support for remote PC-to-host links via four IBM System Network Architectures: interactive 3270, Logical Unit 6.2 Advanced Program-to-Program Communications (LU6.2/APPC), batch-oriented 3770 Remote Job Entry (3770/RJE), and program-to-program Logical Unit 0 (LU0). It can also be used for synchronous PC-to-PC applications via Remote-Talk.

The Adaptcom model C is priced at \$245. AdaptSNA software ranges from \$285 to \$785. For more information, contact **Network Software Associates Inc.**, 22982 Mill Creek, Laguna Hills, CA 92653; (714) 768-4013.

Circle number 17 on the reader service card.

EGA Board Delivers High-Res Image

The Artist 8 is the latest low-cost, high-performance, high-resolution graphics controller board for IBM PCs, XTs, ATs, and compatibles from Control Systems Corporation. Company sources indicate that the Artist 8 delivers the same performance as the Artist 10 at resolutions of 800 x 600 pixels (N) or 1024 x 768 pixels (I). The

board includes Control Systems' Artist GT AutoCAD enhancement interface software. The Artist 8 can be used with both dual-screen and single-screen systems. In single-screen configurations, full-screen EGA can be implemented with a multisync monitor by adding an optional EGA module. The board has 512K of on-board memory and has been designed around the Hitachi ACRTC graphics-control microprocessor and offers 16-colors from a 16.7-million-color palette.

The Artist 8 sells for \$1,295, including the AutoCAD software driver, the Artist GT. For more information, contact **Control Systems Corporation**, 2675 Patton Road, St. Paul, MN 55113; (612) 631-7800.

Circle number 18 on the reader service card.

Board Lets You Plug In To Broader Spectrum

Definicon Systems has begun shipping the GVR-1024 high-resolution board. The board plugs into a socket on Definicon's 68020-based DSI-7xx coprocessor boards, mapping its memory directly on top of the coprocessor's RAM. The GVR-1024 is a three-quarter length board with one megabyte of video RAM, the 82786 controller, a Brooktree chip that permits use of two separate color palettes (256 shades out of 16 million display colors), and a 75-ohm RGB output. The board offers 1024 x 768 resolution in non-interlaced mode. The GVR-1024 also uses dual pixel clocks, the first at 64 MHz to support the main screen and the second at 32 MHz to support the overlay screen. It also uses a 132-pin block to interconnect the graphics card with any DSI-750/780/785 coprocessor board.

The GVR-1024 sells for \$2,499 as a stand-alone unit. DSI-7xx customers can receive an upgrade for

dp_MAX

dBase III Tools in Turbo Pascal 4.0

Complete Support for dBase III files.
DBF, NDX, DBT file & record Access.
Fully Compatible dBase III B+Tree ISAM.
Library of 100+ functions in TP4 Unit.
Allows 250+ files & indexes open at once.
LRU file caching, round robin file manager.

$$dp_MAX = \int_{-\infty}^{+\infty} \frac{dBASE\ III +}{Turbo\ Pascal\ 4.0}$$

Max Software Consultants, Inc.
4101 Greenmount Avenue
Baltimore, MD 21218
(301)-323-5996

\$149

CIRCLE 97 ON READER SERVICE CARD

Version 2.0

XO-Shell™

Pop-Up Utility for Productive People

No matter whether you write programs or reports, **XO-Shell** will help you hurdle the barriers to working faster and more efficiently with memory resident features:

- **VIEW ANY FILE** and **TRANSFER ANY SECTION** into your editor or to your printer
- **DO CROSS-REFERENCING** without leaving your editor
- **VIEW, COPY** and **ERASE** files directly from a **SORTED SCROLLABLE DIRECTORY** Display
- **DO SOURCE-LISTING** while in your application
- **INSERT GRAPHICS CHARACTERS** in your text
- **EXTRACT TEXTS** from data file or wordprocessor generated file
- With a single key stroke **RETRIEVE** previous **DOS** commands, then **EDIT** and **REEXECUTE** them

XO-Shell is for PCs, XTs, ATs, PS/2s, compatibles

"XO-Shell has won its way into my programming toolkit, and I recommend it."
Kent Porter, Dr. Dobbs' Journal

\$79

plus \$5 shipping & handling

Call today toll-free

(800) 635-5011

WYTE CORPORATION

701 Concord Avenue
Cambridge, MA 02138

In MA: (617) 868-7704
Visa, MasterCard

CIRCLE 59 ON READER SERVICE CARD

Coming in the August Issue of Micro/Systems

UNIX ON THE PC

With articles on:

UNIX versus OS/2

How to optimize UNIX's serial I/O

386 UNIX on the PC – a comparative look at:

- SCO Xenix 3
- Microport System/386
- Interactive Systems 386/ix

Running DOS under UNIX : Microport DOSMerge versus Phoenix VP/IX

**Plus: A look at programming the
8086/8088 to PERFORM!**

*And we will have our regular
columns:*

C Forum

**Scientific Computer User
Database Queries
In the Public Domain**

*Including the return of:
The UNIX File*

Don't miss a single issue of
Micro/Systems Journal,
the best information resource
for the PC systems integrator.

\$50 (to cover shipping and handling to send the coprocessor back to the factory). For more information, contact **Definicon Systems, Inc.**, 1100 Business Center Circle, Newbury Park, CA 91320; (805) 498-0652.

*Circle number 19 on the
reader service card.*

Ethernet Simplified with Board Add-in

Simple-NET Systems has released EtherLAN PLUS, a 10 million bps LAN featuring Simple-NET software and Western Digital's EtherCard PLUS PC adapter board. The turnkey system includes a half-sized add-in card, 25 feet of "thinwire," a proprietary NetBIOS interface, and the SimpleWARE network operating system. No DMA is required because of on-board memory buffering, which eliminates memory channel conflicts. Ether-NET PLUS is compatible with IBM's PC, XT, AT, and PS/2 models and compatibles.

The cost for EtherLAN PLUS is \$799.95 per node. For more information, contact **Simple-NET Systems**, 545 West Lambert Road, Suite A, Brea, CA 92621; (714) 526-5151.

*Circle number 20 on the
reader service card.*

VGA-Compatible Monitor Ideal for CAD/CAM

The 20-inch VGA/20 color monitor is the latest addition to Aydin Controls' line of Patriot monitors. The VGA/20 is fully compatible with all modes of the IBM VGA, including 8514/A, and can provide resolutions up to 1024 × 768 pixels with 256 colors. The Aydin monitor also features a high-contrast, anti-glare screen that is ideal for business presentation graphics and CAD/CAM/CAE applications. It also has refresh rates of 60, 70, and 87 Hz for flicker-free images. The monitor is available with optional tilt-and-swivel base, rack-mount kits,

touch systems, and bonded-panel CRTs.

The VGA/20 is available for \$2445, with discount prices as low as \$1,350. For more information, contact **Aydin Controls**, 414 Commerce Drive, Fort Washington, PA 19034.

*Circle number 21 on the
reader service card.*

TOPS Gets Graphics Muscle with Hercules

The Hercules Network Card Plus is now being shipped by Hercules Computer Technology. This PC add-in board combines Hercules high-resolution graphics, including the Hercules' RamFont, with an Appletalk network port. The monochrome card is available for any business that wants to network PCs to other PCs, Macs, and Sun UNIX-based systems using TOPS network software. The Network Card Plus offers standard 720 × 348 pixel resolution.

The card comes with a standard two-year warranty and retails for \$369. For more information, contact **Hercules Computer Technology**, 921 Parker Street, Berkeley, CA 94710; (415) 540-6000.

*Circle number 22 on the
reader service card.*

CPT Adds AT Workstation

A new modular workstation, the CPT 9286, has been added to the CPT Corporation product line. This workstation offers terminal emulations and expansion slots for AT-compatible interface cards that provide communications to virtually any piece of office equipment or host. The CPT 9286 is based on the 80286 processor and has a software-selectable clock speed of 8 or 12.5 MHz. It includes a 1 Mbyte memory and supports up to two floppy and three half-height internal hard disks. CPT's Bridge Board Controller includes an additional 1.2 Mbyte of RAM and permits the unit to run both DOS and CPT text

All the Tools You Need For Motorola 680X0 From Whitesmiths

Whitesmiths, Ltd. now offers a complete set of 68K Cross Development Tools — specifically designed to work together — for the Motorola 68000 family of microprocessors. You get:

A C CROSS COMPILER

Whitesmiths' C Compilers offer the closest conformance currently available to the draft ANSI C Standard. We've added 68020 and 68881 support, and dramatically optimized code generation, so you can get the code quality you need today with the language you'll need tomorrow.

SUPPORT TOOLS

We have all the extras you need to develop embedded programs. Our powerful object utilities help you link multi-segment programs, build direct and sequential libraries, create load maps and interspersed listings, and talk to dozens of downloaders, emulators, and PROM programmers.

C SOURCE LEVEL DEBUGGING

We have the support you need to debug in terms of C functions, data types, and source lines. You debug what you write, not a lower level language.

A MICSIM SIMULATOR

You can debug your embedded programs right on your development host — our MICSIM Simulator needs no extra hardware. It's like debugging on your favorite emulator, but with no contention for dedicated resources, no download time, and with the symbolic breakpoint and trace control you've always dreamed of having.

AN XA8 CROSS ASSEMBLER

Our macro assembler is both fast and powerful, with support for 68020, 68881, and 68551.

A PASCAL COMPILER

You can program as much as you want in ISO Standard Pascal, or use the powerful extensions we've added to this production quality compiler. And you get complete integration with C and assembly language as well.

Working together, the 68K Cross Development Tools deliver both optimized performance and improved programmer productivity. Best of all, Whitesmiths offers everything you need at a very competitive price. We've been delivering and supporting high quality software development tools since 1978, and we're committed to continually enhancing our product line.

If you develop 68000 programs on a DEC VAX, an IBM PC, or a UNIX workstation, chances are we can save you time and money. For more technical details, call our toll-free number today. We also offer attractive packages for OEMs.



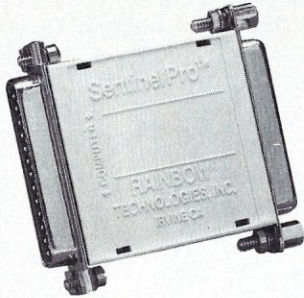
Whitesmiths, Ltd.

59 Power Road
Westford, MA 01886
617/692-7800

CIRCLE 92 ON READER SERVICE CARD

**TOLL-FREE
NUMBER
1-800-225-1030**

Hard Locks for Soft Parts.



At Rainbow Technologies, we think protecting software developers' investments is very serious business. That's why we designed the first fully effective security solution for software running on PCs and other computers.

Our family of virtually impenetrable Software Sentinel hardware keys provides the highest level of software protection the developer can get. While remaining invisible to the end user.

Take a look.

Key Sentinel Family Features.

Prohibits unauthorized use of software □ No need for copy protection □ Unlimited backup copies □ Virtually unbreakable □ Pocketsize key □ Transparent operation □ Transportable

Software Sentinel.

- Runs under DOS and Xenix, on IBM PC/XT/AT and compatibles
- Algorithm technique (Never a fixed response)
- Serial or parallel port version
- Minimal implementation effort
- Higher level language interfaces included
- 100 times faster than fixed-response devices (1ms)

Software Sentinel-C.

- For developers who want to customize or protect multiple packages with one device
- 126 bytes of non-volatile memory that is programmed before shipment of software
- We supply a unique programming adapter for programming the unit

- Higher level language interfaces included
- Runs under DOS on PC/XT/AT and compatibles
- Parallel port version only

Software Sentinel-W.

- Designed for workstations, supermicros and minicomputers
- Serial port only (modem-type)
- Algorithm technique
- We provide detailed interface specifications; Developer creates a port driver
- Interface requirements: 25 pin DB25P or DB25S; RS232/RS422/RS423
- Only signals used: DTR & RTS from computer; signal ground; DSR or optional DCD from Software Sentinel-W or external device. TXD, RXD, CTS, RI passed through.

Call For Software Sentinel Evaluation Kit Pricing.



RAINBOW TECHNOLOGIES

18011-A MITCHELL SOUTH IRVINE, CA 92714 USA
(714) 261-0228 TELEX: 386078 FAX: (714) 261-0260

processing simultaneously.

The base price for the CPT 9286 with half-page monitor, one disk drive, and AT-style keyboard is \$2695. For more information, contact **CPT Corporation**, 8100 Mitchell Road, Minneapolis, MN 55440; (612) 937-8000.

Circle number 23 on the reader service card.

VGA Card Taps Random Access Memory

V-RAM VGA is the latest addition to Video Seven's PC graphics board line. The V-RAM is the first Video Seven board to feature Video Random Access Memory (VRAM). The V-RAM VGA also incorporates FastWrite, a memory-caching mechanism. It also provides BIOS compatibility with the EGA standard, and is backwards compatible with the CGA, MDA, and Hercules standards. It is designed to operate at dot clock rates up to 65 MHz, and offers a variety of on-screen colors (up to 256) and resolutions (1024 × 768 to 640 × 480). Video Seven will bundle software drivers with V-RAM VGA to allow the board to support Microsoft's Windows and Windows/386, and Autodesk's AutoCAD and AutoSHADE.

V-RAM VGA retails for \$799. For more information, contact **Video Seven Inc.**, 46335 Landing Parkway, Fremont, CA 94538; (415) 656-7800.

Circle number 24 on the reader service card.

TOPHAT Tips Backfilled Memory

Boca Research has released a new backfilling memory board for ATs and compatibles, TOPHAT II. The add-in board provides 128K of conventional memory, backfills system memory from 512K to 640K, and operates at CPU speeds up to 10 MHz. The board is both DOS and OS/2 compatible and can connect to any standard 16-bit bus. It also utilizes a 150 ns, 64K × 4 RAM chip, so the TOPHAT II

requires only six RAM chips (compared to the original TOPHAT's 18 64K chips).

TOPHAT II is priced at \$145 in a 128K configuration. For more information, contact **Boca Research**, 6401 Congress Avenue, Boca Raton, FL 33487; (305) 997-6227.

Circle number 25 on the reader service card.

Multiusers Board Taps PS/2 Power

The Multiport/2 is the latest multiuser enhancement product to be developed by Arnet. Designed to take full advantage of IBM's MicroChannel architecture, the Multiport/2 provides eight serial ports that allow connection of terminals, printers, modems and other PCs. The board is compatible with a variety of multiuser operating systems, including Xenix 386 System V, UNIX 386, Pick, PC-MOS/386, Concurrent DOS, BOS, and Theos.

The Multiport/2 retails for \$995. For more information, contact **Arnet Corporation**, 618 Grassmere Park Drive, Suite 6, Nashville, TN 37211; (800) 366-8844.

Circle number 26 on the reader service card. □

Correction

In the May cover story on new trends in "Making DOS Multiuser," we mentioned Interactive Systems's VP/ix, a shell program that enables DOS applications to run under the UNIX multiuser/multi-tasking operating system. Unfortunately, we listed the wrong address for Interactive Systems at the end of the article. The correct address is:

Interactive Systems
2401 Colorado Avenue
Santa Monica, CA 90404
(213) 453-8649

CIRCLE 95 ON READER SERVICE CARD

Break the 640K Barrier for \$59.95.

Introducing Quarterdeck Expanded Memory Managers™: QEMM 386 and QEMM 50/60.

Your 80386 PC, IBM® Personal System/2™ Model 80, PC or AT with 80386 add-in board, as well as your IBM Personal System/2 Models 50 or 60 can all break through the DOS 640K barrier. Now you can have maximum use of your memory—whether you have one megabyte or 32— with the Quarterdeck Expanded Memory Manager. All without having to purchase special expanded memory boards.

Unlock hidden potential beyond 640K.

QEMM uses hidden features within your existing memory to make it compatible with the Lotus-Intel-Micro-soft Expanded Memory Specification (EMS) version 4.0.

Now you can run colossal spreadsheets, databases, and CAD models designed for expanded memory. Using Lotus 1-2-3, Symphony, Framework, Paradox, AutoCAD, Microsoft Excel and more.

Improve productivity with DOS multitasking.

And if you'd like to use these programs all together

—multitasking beyond 640K— QEMM extends the capabilities of our popular DESQview™ multitasking environment.



DESQview lets your programs work together in a familiar way, giving you many of the capabilities and features only promised by other systems. In fact, the editors of *PC Magazine* recently named it "Best Alternative to OS/2," and *InfoWorld* gave it a report card rating of 9.1.

We offer productivity solutions for the forgotten 12 million.

If you are one of the 12 million or so 8088, 8086 or 80286 PC users who feel left out of this new world, don't despair. We have options that let you keep your computer and favorite programs and give you today what the newest computers and operating systems are promising for the future.

Visit your dealer for more information on Quarterdeck products. And ask to see the chart we've prepared explaining today's maze of memory options. and how you can break the 640K barrier.

For 386 PC, IBM PS/2 model 80 and 80386 add-in board users, QEMM/386:

- Fully supports the Lotus-Intel-Microsoft Expanded Memory Specification (EMS) version 4.0
- Enables the 80386 user to configure the 80386's memory beyond the first megabyte as expanded memory or as a combination of both expanded and extended memory.
- Fills out missing DOS memory to 640K and beyond by mapping into unused video memory. When used with a CGA graphics adapter this can give you 736K instead of the standard 640K for DOS.
- Automatically detects the speed of the memory in your PC and uses fast memory whenever possible.
- Increases system performance by remapping slow ROM into the 386's fast RAM memory.
- Frees up DOS conventional memory by loading memory-resident programs above video buffers.
- Supports 386 protected-mode programs, such as Ansa's Paradox 386, that use Phar Lap Software's 386 DOS Extender.
- Turns DESQview into a 80386 control program, using the 80386's 8086 machine architecture to run multiple large programs concurrently.
- Takes approximately 1.5K of overhead from the PC's conventional memory used to run DOS programs.

System Requirements: 80386 PC, IBM PS/2 Model 80, or standard 8088, 8086, 80286 with 80386 add-in board.

For IBM PS/2 model 50 and 60 users, QEMM-50/60:

- Fully supports the Lotus-Intel-Microsoft Expanded Memory Specification (EMS) version 4.0.
- Takes advantage of the hardware capabilities of the IBM PS/2 80286 Memory Expansion Option to transform its "extended" memory into "expanded" memory.
- On PS/2 Model 50 and 60's with 1.5 MB or more of the IBM Memory Expansion Option memory, enables DESQview to run multiple large programs concurrently.

System Requirements: IBM PS/2 model 50 or 60 with IBM PS/2 80286 Memory Expansion Option board or compatible. Note: IBM PS/2 architecture requires disabling motherboard memory and 'backfilling' conventional memory to 1 MB in order for any multitasking environment to work, including Microsoft Windows 2.0, IBM's 3270 workstation program, or DESQview. QEMM will not run on 80286 computers with extended memory.

©1987, Quarterdeck Office Systems. IBM®, Personal System/2™, OS/2™, Lotus®, Intel®, Microsoft®, Excel™, Windows™, 1-2-3®, Symphony™, Framework™, Ansa®, Paradox™, AutoCAD™, PC Magazine®, InfoWorld™, Phar ap™ and AST™ are trademarks of their respective companies.

Rush Me Quarterdeck Productivity Solutions Today!

Product	No. of Copies	Media 3 1/2" or 5 1/4"	Retail Price, ea.	Total
QEMM/386			\$59.95	
QEMM 50/60			\$59.95	
DESQview 2.0			\$129.95	
Shipping & Handling		USA	\$ 5.00	\$
Payment method: <input type="checkbox"/> Check		Outside USA	\$10.00	\$
<input type="checkbox"/> VISA <input type="checkbox"/> MC <input type="checkbox"/> AMEX		Sales Tax (CA residents)	x 6.5%	\$
Valid Since _____ / _____		Expiration _____ / _____	Amount Enclosed	\$

Card Number:

Name on Credit Card:

Shipping Address:

City:

State: Zip: Tel:

Mail to address at right.

Note: If you own DESQview call us for a special upgrade offer, or send in your DESQview registration card. AST Special Edition owners included.

CIRCLE 90 ON READER SERVICE CARD



DOS system running Lotus 1-2-3

SCO XENIX system running SCO Professional

THIS IS AN IBM PS/2 MODEL 80 RUNNING DOS

Under DOS, this PS/2™ is a powerful 80386-based single-tasking, single-user computer that can run thousands of DOS applications. In 16-bit, 8086 mode.

One at a time.

When OS/2™ software becomes available, the PS/2 can become a multitasking, single-user computer running in 16-bit, 286 mode that can also single-task those DOS applications under OS/2.

One at a time.

With DOS or OS/2, the PS/2 will support one user.

	1 user (DOS)	1 user (OS/2)
Cost per system**:	\$12,389	\$12,594
Cost per user:	\$12,389	\$12,594

THIS IS AN IBM PS/2 MODEL 80 RUNNING SCO XENIX

Under SCO XENIX®, this PS/2 becomes a powerful 80386-based multitasking, multiuser computer that can run thousands of XENIX applications. In full-tilt, 32-bit, 386 mode.

Many at a time.

And using SCO VP/ix,™* the PS/2 can multitask DOS applications under SCO XENIX.

Many at a time.

With SCO XENIX, the PS/2 will support one user.

Or 9 users. Or even 33 users.

And it can do all that today because you can get SCO XENIX for the PS/2—*now!*

	1 user	9 users	33 users
Cost per system**:	\$14,559	\$19,726	\$40,402
Cost per user:	\$14,559	\$2,192	\$1,224

SCO XENIX System V and the SCO XENIX family of software solutions are available for all industry-standard 8086-, 80286-, and 80386-based computers, and the IBM® Personal System/2™ Models 50, 60, and 80.



(800) 626-UNIX (626-8649)
(408) 425-7222
FAX: (408) 458-4227
TWX: 910-598-4510 SCO SACZ
uucp: ...decvax!microsoft!sco!info!

*SCO VP/ix available as separate product.

**Cost comparisons are based on most recently published U.S. domestic suggested list prices. Cost model: Base machine: IBM PS/2 Model 80, 70Mb disk, 1Mb RAM, IBM 8512 color monitor, 1Mb additional IBM RAM, IBM ProPrinter XL. 1-user DOS system: Base machine, plus DOS 3.3, WordPerfect 4.2, Lotus® 1-2-3, dBASE III PLUS. 1-user OS/2 system: 1-user DOS system; substitute OS/2 for DOS. 1-user SCO XENIX system: Base machine, plus SCO XENIX 386 for PS/2, SCO VP/ix, SCO Lyrinx® (word processing), SCO FoxBASE+™ (dBASE III PLUS workalike), SCO Professional™ (1-2-3 workalike). 9-user SCO XENIX system: 1-user SCO XENIX system, plus intelligent 8-user multipoint card, 8 IBM 3151 ASCII terminals. 33-user SCO XENIX system: 9-user SCO XENIX system, plus 5 more intelligent 8-user multipoint cards, 24 more IBM 3151 ASCII terminals, 4 Mb additional RAM, additional 70Mb disk.

IBM and ProPrinter XL are registered trademarks and Personal System/2, PS/2 and OS/2 are trademarks of International Business Machines Corporation. • Lotus and 1-2-3 are registered trademarks of Lotus Development Corporation. • dBASE III PLUS is a registered trademark of Ashton-Tate. XENIX is a registered trademark of Microsoft Corporation. • VP/ix is a trademark of INTERACTIVE Systems, Inc. • Lyrinx is a registered trademark and SCO Professional is a trademark of The Santa Cruz Operation, Inc. • FoxBASE+ is a trademark of Fox Software, Inc. 10/87
©1987 The Santa Cruz Operation, Inc., 400 Encinal Street, P.O. Box 1900, Santa Cruz, CA 95061
The Santa Cruz Operation, Ltd., 18 Noel Street, P.O. Box 4YN, London W1A 4YN United Kingdom, +44 1 439 2911, (FAX): +44 1 637 9381, TELEX: 917372 sco us

CIRCLE 94 ON READER SERVICE CARD